NEW ALGORITHMIC METHODS FOR REAL-TIME TRANSPORTATION PROBLEMS

THÈSE N^{O} 2877 (2003)

PRÉSENTÉE À LA FACULTÉ SCIENCES DE BASE

Institut de mathématiques

SECTION DE MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Frank CRITTIN

ingénieur mathématicien diplômé EPF
de nationalité suisse et originaire de Chamoson (VS)

acceptée sur proposition du jury:

Dr M. Bierlaire, directeur de thèse
Prof. M. Ben Akiva, rapporteur
Prof. D. Bonvin, rapporteur
Prof. Th. Liebling, rapporteur
Prof. K. Nagel, rapporteur

Lausanne, EPFL
2004

Acknowledgements

First and foremost, I would like to thank my advisor Michel Bierlaire. This dissertation and the research it documents would not have been possible without his guidance (always consistent and even sometimes anticipatory). His quick intuition was useful in refining the design algorithms. And I want to thank him for the numerous evenings spend in congresses and seminars to bring about scientific progress.

I am indebted to Prof. Thomas Liebling for welcoming me in his group, for sharing his knowledge and wisdom, and for giving me the opportunity to work in a rich environment.

I am also grateful to Prof. Ben-Akiva for having welcome me during three months in his ITS office at the MIT. I want to thank the ITS staffs and students for their warm welcome, especially Costas for his help in making my non-scientific experience at MIT a great one.

I wish to thank the other members of my thesis committee, the president Prof. Dacorogna, Prof. D. Bonvin and Prof. K. Nagel for their constructive commentaries.

This thesis has been supported by Swiss National Science Foundation grant number 2100-63848.00

I would also like to thank all my colleagues of the ROSO, with an honorary mention to my office roommate Michela for her patience to answer my incessant linuxo-latexo-computer questions, Michael "Schumi" for reading in detail this report, Jean-Albert for his calm even during the search of our luggage and all the others for providing a friendly and creative atmosphere.

I would like to thank Yannik for being a cool roommate, a though opponent and mainly a friend during these years of study, with a wink to the "ragondins".

Finally, I would like to thank my parents to whom I dedicate this work, and of course Sylvie for whom I keep my charming compliments.

Abstract

Two of the most basic problems encountered in numerical optimization are least-squares problems and systems of nonlinear equations. The use of more and more complex simulation tools on high performance computers requires solving problems involving an increasingly large number of variables. The main thrust of this thesis the design of new algorithmic methods for solving large-scale instances of these two problems. Although they are relevant in many different applications, we concentrate specifically on real applications encountered in the context of Intelligent Transportation Systems to illustrate their performances.

First we propose a new approach for the estimation and prediction of Origin-Destination tables. This problem is usually solved using a Kalman filter approach, which refers to both formulation and resolution algorithm. We prefer to consider a explicit least-squares formulation. It offers convenient and flexible algorithms especially designed to solve large-scale problems. Numerical results provide evidence that this approach requires significantly less computation effort than the Kalman filter algorithm. Moreover it allows to consider larger problems, likely to occur in real applications.

Second a new class of quasi-Newton methods for solving systems of nonlinear equations is presented. The main idea is to generalize classical methods by building a model using more than two previous iterates. We use a least-squares approach to calibrate this model, as exact interpolation requires a fixed number of iterates, and may be numerically problematic. Based on classical assumptions we give a proof of local convergence of this class of methods. Computational comparisons with standard quasi-Newton methods highlight substantial improvements in terms of robustness and number of function evaluations.

We derive from this class of methods a matrix-free algorithm designed to solve large-scale systems of nonlinear equations without assuming any particular structure on the problems. We have successfully tried out the method on problems with up to one million variables. Computational experiments on standard problems show that this algorithm outperforms classical large-scale quasi-Newton methods in terms of efficiency and robustness. Moreover, its numerical performances are similar to Newton-Krylov methods, currently considered as the best to solve large-scale systems of equations. In addition, we provide numerical evidence of the superiority of our method for solving noisy systems of nonlinear equations.

This method is then applied to the consistent anticipatory route guidance generation. Route guidance refers to information provided to travelers in an attempt to facilitate their decisions relative to departure time, travel mode and route. We are specifically interested in consistent anticipatory route guidance, in which real-time traffic measurements are used to make short-term predictions, involving complex simulation tools, of future traffic conditions. These predictions are the basis of the guidance information that is provided to users. By consistent, we mean that the anticipated traffic conditions used to generate the guidance must be similar to the traffic conditions that the travelers are going to experience on the network. The problem is tricky because, contrarily to weather forecast where the real system under consideration is not affected by information provision, the very fact of providing travel information may modify the future traffic conditions and, therefore, invalidate the prediction that has been used to generate it.

Bottom (2000) has proposed a general fixed point formulation of this problem with the following characteristics. First, as guidance generation involves considerable amounts of computation, this fixed point problem must be solved quickly and accurately enough for the results to be timely and of use to drivers. Secondly the unavailability of a closed-form objective function and the presence of noise due to the use of simulation tools prevent from using classical algorithms. A number of simulation experiments based on two system softwares including DynaMIT a state-of-the art, real-time computer system for traffic estimation and prediction, developed at the Intelligent Transportation Systems Program of the Massachusetts Institute of Technology (MIT), have been run. These numerical results underline the good behavior of our large-scale method compared to classical fixed point methods for solving the consistent anticipatory route guidance problem.

We close with some comments about future promising directions of research.

Résumé

La résolution de moindres carrés et de systèmes d'équations non-linéaires font parties des problèmes fondamentaux que l'on rencontre en optimisation numérique. De part les progrès constants réalisés en informatique favorisant l'utilisation d'outils de simulation de plus en plus complexes, la taille de ces problèmes ne cesse d'augmenter. L'objectif principal de cette thèse est la conception de nouvelles méthodes algorithmiques pour la résolution de ce type de problèmes de grande taille. Bien qu'ils apparaissent dans de nombreux domaines, cette dissertation se concentre sur l'application de nos méthodes à des outils de gestion dynamique du trafic routier.

Une nouvelle approche pour l'estimation et la prédiction de tables Origine-Destination est proposée. En général ce problème se résout avec la méthode des filtres de Kalman, qui fait référence aussi bien à la formulation qu'à l'algorithme de résolution. Nous préférons ici une formulation en termes de moindres carrés. Elle permet l'utilisation d'algorithmes efficaces pour la résolution de problèmes de grande taille. Les résultats numériques mettent en évidence l'amélioration flagrante en termes de temps de calculs de cette approche par rapport à l'algorithme des filtres de Kalman. De plus, cette méthode permet de considérer des problèmes de plus grandes dimensions susceptibles de se présenter dans les applications réelles.

Une nouvelle classe de méthodes pour la résolution de systèmes d'équations non linéaires est présentée dans cette thèse. L'idée principale est la généralisation des méthodes sécantes, en construisant un modèle basé sur plus de deux itérés. Nous proposons une approche basée sur les moindres carrés pour la calibration du modèle, contrairement aux méthodes traditionnelles qui utilisent l'interpolation. Sous des hypothèses classiques, la convergence locale de cette classe de méthodes est démontrée. Les résultats numériques obtenus par ces algorithmes comparés aux méthodes quasi-Newton illustrent une amélioration significative des performances aussi bien du point de vue de l'efficacité que de la robustesse.

A partir de cette nouvelle classe de méthodes, nous développons un algorithme pour la résolution de systèmes d'équations de grande taille qui ne recourt pas à la construction explicite de matrices et qui n'impose aucune structure particulière au problème. En pratique cet algorithme nous a permis de traiter des problèmes comptant un million de variables. Les résultats numériques

obtenus sur des problèmes standards montrent clairement que cette méthode est supérieure aux méthodes quasi-Newton classiques en termes d'efficacité et de robustesse. Ces performances peuvent même être comparées avec les méthodes de type Newton-Krylov actuellement considérées comme les meilleures pour la résolution de systèmes d'équations de grande taille. De plus, la qualité de notre algorithme appliqué à des problèmes pour lesquels l'évaluation de la fonction objectif est bruitée est mis en évidence.

Cette méthode est ensuite appliquée au problème de gé nération d'information routière cohérente. Nous nous intéressons en particulier à la génération en temps réel d'information, basée sur la prédiction de l'état futur du réseau routier. Dans ce cas, il faut tenir compte de la réaction des conducteurs à cette information. En effet, contrairement aux informations météorologiques, la diffusion d'informations sur le trafic influence celui-ci, affectant les prédictions. Le concept de cohérence entre l'information produite et les prédictions effectuées appréhende ce problème. La génération d'information routière cohérente peut être formulée comme un problème de point fixe avec les caractéristiques suivantes. Premièrement, comme la génération de l'information se base sur des simulations de l'état futur du réseau qui nécessitent beaucoup de temps de calcul, le problème de point fixe doit être résolu rapidement afin de pouvoir disséminer ces informations à temps. Deuxièmement, l'utilisation de modèles comportementaux afin de simuler la réaction des usagers empêche l'application de méthodes classiques souvent sensibles à la présence de bruit dans la fonction objectif. Les tests numériques sont effectués grâce à deux programmes. L'un d'entre eux est DynaMIT, un système de prédiction de trafic et de génération d'information en temps réel développé au Intelligent Transportation Systems (ITS) Program du Massachusetts Institute of Technology (MIT). Ces résultats numériques mettent en évidence l'excellent comportement de notre méthode comparé aux méthodes de résolutions conventionnelles.

La thèse se termine par l'identification de directions prometteuses qui s'inscrivent dans le prolongement des recherches présentées.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 General framework of the research

A model can be defined as a simplified representation of reality. Recently, following the continuous progress realized in computer science, and although these models are still simplified representations, the use of computer simulations of more and more complex models is growing in importance. Consequently the most basic problems encountered in numerical optimization involve an increasingly large number of variables, where existing methods for resolving such problems are often unsuitable.

This thesis focuses on the design of new algorithmic methods for solving two fundamental problems of numerical optimization:

- resolution of least-squares problems,
- resolution of systems of nonlinear equations.

Their widespread applicability, in engineering, economics, and industry, motivates the need for developing and studying efficient large-scale algorithms for solving them. In this dissertation, we concentrate specifically on real applications often encountered in the context of transportation. These applications involve complex simulation tools and are associated with large-scale real-time problems providing a stimulating test bed environment for the proposed algorithms.

1.2 Intelligent Transportation Systems

When analyzing transportation problems, proposing effective traffic congestion reduction strategies, and designing powerful algorithms for solving complex mathematical problems, researchers encounter the same difficulties. For both traffic engineers and mathematical researchers, there is more than one approach to a particular problem. The best solution may not necessarily be the first one that comes to mind, so finding efficient resolution strategies may require looking at the problem in different ways. Problem analysis is by nature an iterative process: an analyst perceives a problem, conceptualizes an approach, tries it out and revises to better fit the problem.

First illustration of this observation is given by the historical answer given to traffic congestion. The need for transportation, particularly in metropolitan areas, continues to grow as population increases, inducing more and more traffic congestion. Traffic jams cause considerable costs due to unproductive time loss. They also increase the possibility of accidents, plus have a negative impact on the environment (air pollution, lost fuel), and decrease the quality of life (noise, stress). There are many possible ways to reduce congestion. In the past, transportation problems were usually evaluated in terms of supply and solved by building new infrastructures, such as highways, roads and parking facilities. Unfortunately the price to pay for this is brute force transportation

policy. This results in a number of undesirable induced problems such as pollution, increased accident rates, and inefficient use of transportation systems. The lack of readily available land on which to build, further complicates matters, plus it is a less attractive option due to increasing construction costs and legitimate environmental concerns. These problems have influenced most countries to change from infrastructure-intensive strategies towards more balanced and sustainable transportation solutions. Among these solutions, Intelligent Transportation Systems (ITS) are systems that apply emerging information technologies and innovative approaches to improve general traffic conditions (see Transportation Research Board, 1999). ITS have received a tremendous amount of attention in the last decade, both from practitioners and researchers.

When considering the economic interpretation of transportation, the interaction between demand and supply is a basic concept which is fundamental for understanding, designing, and managing intelligent transportation systems. The demand side of transportation tries to answer a number of questions. Who is traveling? From where to where? When do those trips begin and end? What routes or modes are they taking? Transportation demand includes travel demand, route choice, and responses to information. The supply side of transportation includes the physical network, the traffic control devices, and all the incidents and events occurring within the network. In summary, the supply characterized by the network and anything that affects its capacity. One of the most important is the congestion, which arises when demand levels approach the total capacity of a facility while the time required increases well above the average journey time for low demand conditions.

In the context of intelligent transportation systems, the main role of Dynamic Traffic Management Systems (DTMS) is to improve general traffic conditions using advanced technologies managed by real-time intelligent software systems. A classical framework of DTMS generally combines three elements: advanced surveillance systems collecting real-time traffic data, Advanced Traffic Management Systems (ATMS) and Advanced Traveler Information Systems (ATIS). The surveillance systems are mainly composed of traffic sensors and probe vehicles. The ATMS affect the network supply, imposing restrictions and constraints on traffic flows, predicting traffic congestion, and providing alternative routing instructions to vehicles over wide areas using, for example, signal control and ramp metering to improve the network performance. The ATIS affects the transport demand by providing historical, real-time, and predictive information to support drivers travel decisions before and during their trip. Its aim is to reduce overall delays and improve travel time reliability. ATIS support includes radio forecasts, web-based or on-board GPS-based navigation systems, and variable message signs. Such systems differ from ATMS in that drivers are not forced to follow the systems recommendations.

An effective application of these systems (ATMS/ATIS) using DTMS must be based on an implicit or explicit simulation of the interaction between demand and supply. DynaMIT (DYnamic Network Assignment for the Management of

Information to Travelers) is such a system, designed for use in traffic management centers. It is a simulation-based real-time system that estimates the current state of a transportation network, predicts future traffic conditions, and provides consistent and unbiased information to travelers. It uses a demand and supply simulator to perform these tasks. DynaMIT uses past data, real-time surveillance, and estimated traffic control settings to generate prediction-based guidance with respect to departure time, pre-trip route, mode choice decisions, and en-route choice decisions. A brief description of this system is given in Chapter 2.

The emergence of Intelligent Transportation Systems have considerably affected the field of transportation modeling. The complexity of these systems requires the development of sophisticated tools, such as DynaMIT, for optimal system exploitation. The systems potential requires transportation modelers to describe exactly the interaction between travelers, DTMS, and supply/demand. New models and new methodologies have been proposed, creating new mathematical challenges. These challenges are the motivation for the research presented in this thesis.

1.3 Thesis applications

This thesis proposes new formulations and solution algorithms to efficiently deal with two issues constituting the backbone of dynamic traffic management systems:

- Origin-Destination tables estimation and prediction
- Consistent anticipatory route guidance generation

The first subject is at the frontier between demand and supply represented by the dynamic Origin-Destination (OD) tables. They represent the amount of demand per origin, destination and departure time. OD tables capture transportation demand at an intermediate level of aggregation, sufficiently disaggregated to provide input data for the dynamic traffic assignment models (responsible for assigning the demand on to the network), and sufficiently aggregated to have interesting statistical properties. This thesis proposes a new method for the real-time estimation and prediction of dynamic OD tables designed for large-scale applications. This method is based on modeling assumptions proposed by Ashok (1996), and is designed to better exploit the mathematical structure of the problem.

The second subject is the consistent anticipatory route guidance (CARG) generation, which presents an even greater challenge. It consists of using available data from the surveillance system in order to generate information disseminated by the ATIS.

There are several categories of travel information. Information can be provided before the trip starts (pre-trip) or during the trip (en-route). It can

describe available trip options (descriptive) or provide specific travel recommendations (prescriptive), or a combination of the two. It can be generated from past and current traffic conditions (reactive), or on future traffic conditions (proactive or anticipatory). Following Bottom (2000), in this dissertation, the terms *guidance* and *information* will be used interchangeably and will refer to any prescriptive and/or descriptive messages given to the drivers.

The need here is to generate consistent anticipatory information, both pre-trip and en-route, descriptive or prescriptive. Consistent means that the anticipated traffic conditions used to generate the information must be similar to the traffic conditions that the travelers are actually going to experience on the network. The problem is significant because, contrary to weather forecasting where provided information does not affect the real system, the very fact of providing travel information may modify the future traffic conditions and therefore invalidate the prediction that has been used to generate it. Consequently, travelers responses to information must be explicitly captured in the generation process. If not, an overreaction phenomena may be observed (Ben-Akiva et al., 1996) and the information systems reliability will be affected.

Bottom et al. (1999) and Bottom (2000) have proposed a formal description of the consistent anticipatory route guidance problem, using a detailed analysis framework. It is formulated as a fixed point problem. For a large network, this fixed point problem involves a high number of variables that produce high-dimension problems. Moreover the complexity of transportation systems and the necessity of capturing traveler behavior impose the use of disaggregated models and simulation-based tools, consequently the mapping which defines the fixed point is non-analytical and stochastic. We refer to Bottom (2000) for a detailed analysis of the sources of stochasticity. In this thesis the approach to address the effects of stochasticity in the computation of the fixed point is to threat simulator outputs not as a realization of a stochastic process but rather as a time trajectory of deterministic values that are affected by noise.

Bottom (2000) has identified in the literature suitable fixed point algorithms to solve this problem. Preliminary numerical tests have illustrated that the slow behavior of these algorithms considerably limits their use for real-time traffic management systems. This thesis proposes to consider the route guidance generation problem as the resolution of systems of nonlinear equations which are formally equivalent to fixed point problems. This means that a new class of algorithms have been developed for solving nonlinear equations, which will overcome the complexity of these problems due to its distinctive features in the context of the consistent anticipatory route guidance generation problem.

The methodological solutions designed to address the problems of both OD estimation and prediction, and consistent anticipatory route guidance generation, share the same characteristics. The large-scale characteristics is common to from these two problems due to the necessity of using detailed network representations and disaggregate models for producing reliable travel behavior and demand. The real-time limitation in producing an OD table or a consistent

guidance is inherent in DTMS. This difficulty is accentuated by the large-scale character of the problem and the sophisticated simulation tools involved in the computation of these values.

As for the applications themselves, the first methods that were proposed have been derived directly from the original formulations for these two problems. Unfortunately, due to intrinsic constraints, they are not suitable for real applications. In this thesis, new formulations are defined for these problems, which allow different approaches to be considered as solutions.

Despite the fact that the applications described are essentially in transportation field, we are convinced that the presented approaches have a wide range of other applications in many domains.

1.4 Thesis contributions

We see six important contributions of this thesis

- The development of a new formulation of the dynamic OD estimation and prediction problem, based on a combination of the approaches given by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993). The proposed least-squares formulation enables to consider the most efficient algorithms for exploiting the sparsity of involved matrices. This new formulation and its associated solution algorithm provide an efficient approach to solve large-scale OD estimation and prediction in real time.

- The solutions proposed to solve large-scale incremental least squares are applicable to the specific OD tables estimation and prediction problem, and can also be applied to any model that aims to design and efficiently solve sparse and large-scale problems with a Kalman filter approach.

- The creation of a new class of methods for solving systems of nonlinear equations. This class of methods is the first effective multi-iterate generalization of classical secant methods. It results in a significant improvement in terms of robustness and number of function evaluations as compared to classical quasi-Newton methods. Moreover, these methods are robust in the presence of noisy systems.

- A matrix-free algorithm has been designed to solve large-scale systems of nonlinear equations without any particular assumption about the structure of the problem or the Jacobian. Its numerical performances are similar to Newton-Krylov Methods, which are currently considered as the best algorithms to solve large-scale systems of nonlinear equations. Moreover our approach exhibits robust behavior in the presence of noise in the equations, emphasizing the added value of our approach compared to Newton-Krylov methods.

- So far, no satisfactory solution algorithm has been proposed for the consistent anticipatory guidance generation problem. The work in this thesis, as a direct continuation of the innovative work by Bottom (2000), has investigated approaches never before considered in this context, and proposes to improve them according to the problems specific characteristics.

- The results of this methodological and algorithmic research are applicable and have been successfully applied in the context of real-time predictive information systems like DynaMIT.

1.5 Thesis organization

In order to precisely characterize both applications of this work, the components of a dynamic traffic assignment system (DynaMIT) are briefly described in Chapter 2. It is a simulation-based real-time system with traffic prediction capabilities providing anticipatory route guidance. The general framework of the system is given using its two main components: the demand simulator and the traffic simulator. The requirements deriving from the design of such a system are emphasized, both for the resolution of the dynamic OD estimation and prediction and the CARG problem.

Chapter 3 is dedicated to the demand side. An approach is presented for estimating and predicting Origin-Destination (OD) tables. In the specific context of DTMS, the dynamic nature of the problem and the real-time requirements make it intricate. A least-squares modeling approach for solving the OD estimation and prediction problem is considered, which seems to offer convenient, flexible algorithms such as LSQR, an iterative algorithm proposed by Paige and Saunders (1982) for solving large-scale least squares. The formulation presented here is inspired by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993). The dynamic nature of the problem is represented by an auto-regressive process, which captures the serial correlations of the state variables.

The standard Kalman filter algorithm is compared to LSQR, that explicitly exploits matrix sparsity, allowing the consideration of larger problems that are likely to occur in real applications. The LSQR algorithm is shown to significantly decrease the computation effort needed by the Kalman filter approach for large-scale problems. A theoretical number of flops are provided for both algorithms, in order to predict which algorithm will perform better on a specific instance of the problem. Chapter 3 contains mainly the developments presented in Bierlaire and Crittin, 2004. This paper has been accepted by *Operations Research*.

Chapter 4 and Chapter 5 propose a new class of methods for solving systems of nonlinear equations. This part of the dissertation is independent of the transportation applications. The principal objective is to generalize classical secant methods by building a linear model of the objective function by using more than two previous iterates. This approach allows all the available infor-

mation collected through the iterations to be used for constructing the model. A least-squares approach is preferred for calibrating the linear model, as exact interpolation requires a fixed number of iterates, and may be numerically problematic.

In Chapter 4, an explicit control of the numerical stability of the method is proposed. This approach leads to a Broyden style update formula. In this case, we prove the local convergence of the corresponding quasi-Newton method. Computational comparisons with classical methods, based on performance profiles, highlight a significant improvement in term of robustness and number of function evaluations compared with classical secant methods. Preliminary numerical tests are presented showing the robust behavior of the new methods in the presence of noisy system of nonlinear equations.

Chapter 5 introduces a matrix-free algorithm derived from these class of methods, which is designed to solve large-scale systems of nonlinear equations without any particular assumption about the structure of the problem or its Jacobian. Extensive computational experiments on standard problems show that this algorithm outperforms classical large-scale quasi-Newton methods in terms of efficiency and robustness. Its numerical performances can even be compared to the well-known Newton-Krylov methods currently considered as the best algorithm for solving large-scale systems of nonlinear equations. Numerical results with noisy systems clearly demonstrate the superiority of the population-based approach in presence of stochasticity.

Chapter 4 and Chapter 5 are constituted for the most part of ideas presented in Bierlaire and Crittin, 2003a and Bierlaire and Crittin, 2003b. This last paper has been submitted to *Mathematical Programming*.

Chapter 6 is dedicated to solving the problem of consistent anticipatory route guidance generation. Bottom (2000) has proposed a general fixed point formulation of this problem with the following characteristics. First, as guidance generation involves considerable amounts of computation, this fixed point problem must be solved quickly and accurately enough for the results to be timely delivered to drivers. Second, the unavailability of analytical forms for the objective function and the presence of noise due to the use of simulation tools prevent the use of classical algorithms. A new formulation of this problem is given as the resolution of a system of nonlinear equations and uses the large-scale method proposed in Chapter 5 in order to handle the intrinsic characteristics of the consistent anticipatory route guidance generation, especially the high dimension associated with real problems.

Finally, a number of simulation experiments are presented, they are based on two different simulation tools in order to compare the performances of the diverse algorithms. The first is a simple simulator implementing the framework of the route guidance generation on a small network, which is used to illustrate the properties of this problem and the behavior of the algorithms. Then, case studies are presented, based on different networks using DynaMIT, including an instance containing more that 120'000 variables. These results point out

the real-time potential of the method as its ability to handle large-scale problems. They also underline the good behavior of this new proposed method as compared to fixed point methods for solving the CARG problem. Chapter 6 is an extension of the concepts and the preliminary results presented in the two papers: Bierlaire and Crittin, 2003c and Bierlaire and Crittin, 2001.

The conclusions are presented in Chapter 7. They provide a summary of the main results presented and an inventory of promising directions for future research that are related to the subjects of this dissertation or came to light because of it.

Chapter 2

DynaMIT

Contents

2.1 General framework

DynaMIT is a state-of-the-art, real-time computer system for traffic estimation and prediction with route guidance generation capabilities. DynaMIT is the result of 10 years of intense research and development at the Intelligent Transportation Systems Program of the Massachusetts Institute of Technology (for description and details, see Ben-Akiva et al., 2002, Bottom et al., 1999 and Ben-Akiva et al., 1998). DynaMIT is designed to operate in real time, using traffic volume and control system state data to estimate and predict time-dependent origin-destination flows and network conditions, while generating descriptive and prescriptive information that should be consistent with the predicted traffic conditions. Consistency ensures that the predictions of expected network conditions, obtained by the system, match what drivers would experience on the network.

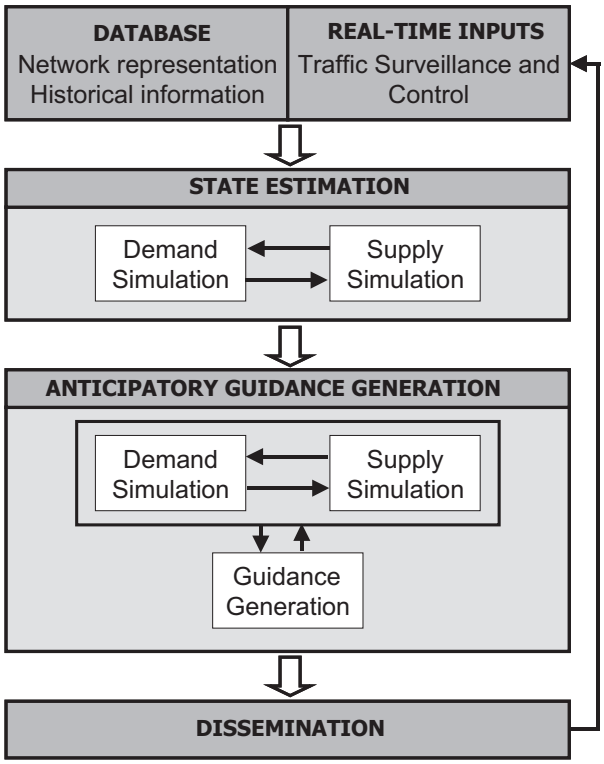


Figure 2.1: General Framework of DynaMIT

DymaMIT is composed of various models and algorithms designed to estimate and predict time-dependent OD flows and network conditions, and to generate prediction-based information for a given time horizon. The state estimation is based on a simulation of the interaction between network supply and traveler demand. It evaluates the current state of the network based on

historical (historical OD flows) and real-time data (traffic sensors, control logic of traffic signals, probe vehicles or mobile phones). The prediction-based information generation uses the current state of the network as input and generates anticipatory route guidance also based on supply-demand interaction.

As described in the introduction, these two main functionalities are accomplished by the interaction between a supply and demand simulator (Ben-Akiva et al., 2002, Bierlaire et al., 2000, Antoniou et al., 1997, Ben-Akiva et al., 1997), as illustrated by Figure 2.1. The demand simulator estimates and predicts OD flows, characterizes pre-trip decisions such as departure times, mode choices and path choices for each network user. It also determines the en-route decisions defined as path switching strategies in response to information. As the demand simulator has to explicitly simulate the choices of each driver, it includes disaggregated behavioral models (Antoniou et al., 1997). The supply simulator evaluates the impact of the demand on the network based on the transportation network and the traffic control system (Ben-Akiva et al., 1997). It uses a mesoscopic traffic simulator, which includes a representation of the topology of the network, with the explicit traffic dynamics of congestion development and dissipation. The supply simulator is also responsible for assigning the OD flows to the link flows at sensor locations.

2.2 State estimation

The main function of the state estimation is to generate a complete description of the current state of the network that is coherent with the real-time information. To perform this estimation, DynaMIT uses historical OD tables, travel time information disseminated to the users and real-time traffic flows collected from the surveillance system.

Practically speaking, DynaMIT estimates the initial demand, then updates historical OD tables by explicitly simulating the reaction of individuals to guidance information relying on behavioral models. The historical demand changes are then aggregated to obtain updated historical OD tables. At this point, the corresponding updated OD flows may not accurately match the real-time measurements of actual link flows on the network. Because of this possible divergence, the OD flows have to be adjusted in real time to fit the surveillance data. This problem, often called dynamic real-time estimation and prediction of time-dependent OD flows, is a complex under-determined least-squares problems with real-time requirements.

The formulation and resolution of this problem in DynaMIT is currently based on a Kalman Filtering framework. It solves the associated least squares in an incremental fashion, allowing the solution to be updated when additional data are available. Unfortunately, this procedure is not really designed to handle large-scale problems. A detailed description of the problem, as well as a new approach for solving this problem in large scale, is given in Chapter 3.

After these adjusted OD flows are computed, DynaMIT simulates traffic conditions using the given traffic control strategies and guidance information disseminated to users. Behavioral models are then used to capture the response of users to ATIS in the form of en-route path choices.

The OD estimation and prediction process involves assignment matrices, which represent the mapping of OD flows into link flows computed by the supply simulator, which itself uses OD matrices as inputs. Consequently, as illustrated by Figure 2.1, a fixed point problem has to be solved at each time interval, in order to find the description of the demand that is congruent with the supply. This fixed point is approximated simply by iterating, meaning that the estimation and prediction of OD tables has to be performed several times for each time interval reinforcing the use of an efficient method for computing the OD tables.

The outcome of the state estimation phase is an estimation of the actual traffic conditions on the network, which will serve as the basis for guidance generation.

2.3 Anticipatory guidance generation

Guidance refers to the information provided to travelers in an attempt to facilitate their decisions relative to departure time, travel mode, and route. Route guidance is said to be anticipatory or proactive if it is based on future traffic conditions, as opposed to reactive guidance, which is based on historical or current traffic conditions.

Providing traffic information based on predicted traffic conditions may produce a phenomenon called overreaction. This happens when the number of drivers, changing their behavior in response to the information, is so high that the predicted traffic conditions used as a basis for the guidance generation becomes invalid. Such a phenomenon is undesirable, because it jeopardizes the effectiveness and reliability of the information system.

In order to avoid this problem, the guidance generation process must explicitly anticipate the driver's reactions to the information. As a consequence, a self-referenced problem is obtained, where the generated information depends on the predicted traffic conditions, that in turn depends on the generated information. Bottom et al. (1999) and Bottom (2000) proposed a new general formulation of the problem (described in details in Chapter 6). Based on three sets of state variables, the problem is posed as a fixed point problem $T(x) = x$, where x is one of the three sets of state variables. T is the combination of three mappings, involving traffic simulation, behavioral model, and traffic information dissemination.

In the context of DynaMIT, the mapping evaluation T is simulated by the interaction between the supply and demand simulator. It is done in the same way as the network state estimation except for the future OD tables, where an

autoregressive process is used as the prediction tool. Based on the resulting prediction, a new guidance is generated using a fixed point method. This guidance is then evaluated using supply and demand simulation.

Currently, the algorithm implemented in DynaMIT is a fixed point method based on the Banach contraction principle. It is called the time smoothing algorithm. This procedure, described in Chapter 6, is inspired by the method of successive averages (MSA), a recursive averaging method designed to address noisy problems. Unfortunately, these algorithms exhibit slow behavior, a problem which motivates the design of new methods to solve this critical fixed point problem in the DynaMIT system.

2.4 Conclusion

DynaMIT is a simulation-based system designed to generate, in real time, consistent anticipatory route guidance. To estimate the current state of the network and produce predictions of future traffic conditions, DynaMIT uses a supply and demand simulator. The state estimation and prediction requires several estimations and predictions of dynamic OD tables, where an iterative algorithm is used to reach congruence between demand and supply. Hence for real-time applications, the method used to estimate and predict these OD tables has to be efficient in terms of computation time. The guidance generation process involves the resolution of a fixed point problem in order to produce consistent anticipatory route guidance. One evaluation of the mapping, defining this fixed point problem, involves the supply and demand simulator consuming a large amount of computational time. Therefore the design of an efficient method to solve this fixed point problem is unavoidable for producing a system with real-time capabilities.

Chapter 3

OD tables

Contents

3.1 Introduction

As described in the introduction , with regard to the demand aspects, the estimation and prediction of OD tables has become an important element of Dynamic Traffic Management Systems (DTMS) (Ashok and Ben-Akiva, 1993, Bierlaire et al., 2000, Ben-Akiva et al., 2002). The main difficulty of the problem is due to the following characteristics:

1. The dynamic nature of the process must be captured in the modeling framework.

2. Only indirect measurements of OD flows can be obtained through link flows. Therefore, the estimation problem is intrinsically under-determined for non trivial problems, as there are usually more unknowns than the number of observations.

3. Due to real-time requirements of DTMS, current and future OD flows must be available at any point in time, based on the most up-to-date data. Then, as time proceeds and more data becomes available, the solution must be updated to reflect the evolution of the network conditions.

Several approaches have been proposed in the literature to model the dynamic nature of demand. van der Zijpp (1996) proposes an approach based on time-space trajectories, Chang and Wu (1994) use a random walk model, Okutani (1987) describes the dynamics through an auto-regressive formulation capturing serial formulation across OD flows of subsequent time intervals. Ashok and Ben-Akiva (1993) propose an auto-regressive process as well, but based on the deviations between actual and historical OD flows.

Overcoming the under-determination of the estimation problem has also been approached in various ways. For static OD estimation, concepts like gravity (Casey, 1955), entropy (Wilson, 1970, Willumsen, 1981) and information theory (Van Zuylen and Willumsen, 1980) have been proposed. However, the use of an a priori OD table, derived from surveys or from previous studies is the most common way to overcome the under-determination. For dynamic OD estimation, the a priori table may be obtained from historical database, from a one-step prediction of a table estimated for the previous time-interval, or even using probe vehicles data (see Ashok, 1996 for more details).

The Kalman filter algorithm (Kalman, 1960) has been widely proposed to accommodate the real-time requirements (Okutani and Stephanades, 1984, Ashok and Ben-Akiva, 1993, Ashok and Ben-Akiva, 2000, Chang and Wu, 1994, van der Zijpp and Hammerslag, 1994). This algorithm solves a least-squares problem in an incremental fashion, allowing to update the solution when additional data is available.

In this chapter, we derive a least-squares model, combining the formulation proposed by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993). The state variables are the deviations between historical and actual OD flows. The main

motivation is to indirectly take into account all experiences gained over many prior estimations, and accumulated in the historical data. Moreover it also gives statistical stability as the deviations can be more realistically assumed to be normally distributed with zero mean.

The Kalman filter algorithm has been implemented in the DynaMIT system (Antoniou et al., 1997, Ben-Akiva et al., 2002, Ben-Akiva et al., 2001). The main drawback of the Kalman filter algorithm appears to be its inability to handle large-scale problems. Indeed, even if efficient implementations are used (Chui and Chen, 1991), the analytical computation of the normal equations and the variance propagation require intensive linear algebra computation. Moreover, the sparsity of the least-squares problem is not exploited by the algorithm, and a lot of fill-in is taking place. Another limitation of the Kalman filter is its constant numerical complexity. When traffic conditions are normal, or when the time intervals are short, the auto-regressive process can provide a pretty accurate estimate of the OD table. The Kalman filter algorithm always consumes the same amount of computational resources, irrespectively of the quality of the a priori matrix.

It is important to make the distinction between the *model formulation* and the *solution algorithm*. Usually, the model formulation is motivated by the use of the Kalman filter algorithm (e.g. Ashok, 1996), and the name *Kalman filter* refers to both the model and the algorithm. We consider Kalman filtering as an incremental algorithm to solve a least-squares problem in a real-time context (Bertsekas, 1995). The use of a least-squares approach to solve the dynamic OD estimation problem has been originally proposed by Cascetta et al. (1993). In this chapter, we build on their modeling framework by (i) exploiting Ashok and Ben-Akiva, 1993 proposal of using deviations as state variables, and an auto-regressive model combined with historical data to obtain an a priori OD table, and (ii) providing an efficient algorithm to solve the problem in real time.

We propose to use the LSQR algorithm (Paige and Saunders, 1982), analytically equivalent to a conjugate gradient method, requiring only matrix-vector products and, therefore, explicitly accounting for the problem's sparsity. In order to avoid to compute the variance propagation, which produces a great deal of fill-in in the matrices, all OD tables, for all time intervals within the considered horizon, must be included in the state vector. The associated model therefore grows with time, and may become intractable. This is not acceptable for systems supposed to run continuously in time, *i.e.*with a virtually infinite horizon. Therefore, we include only a limited number of past time intervals in the estimation process. We assume that the estimators and associated variance-covariance matrices for previous time intervals are given. This assumption is reasonable for all practical purposes, as the impact of new data on old OD tables becomes insignificant with time.

In theory, LSQR converges in n iterations, where n is the number of variables to estimate. In the case where the actual deviations are sufficiently well predicted by the auto-regressive process, the iterative nature of LSQR makes

it converge in a few iterations, significantly decreasing the computational burden. The Kalman filter algorithm, based on a direct method, has a constant computational cost and, therefore, does not exploit such advantages.

This chapter is organized as follows. The real-time traffic prediction is described in Section 2 with its new least-squares formulation. Section 3 presents the two solution algorithms considered describing their theoretical advantages and weakness. The operational applications of both algorithms are discussed in section 4. Numerical results are presented in section 5 and some conclusions and perspectives are outlined in Section 6.

3.2 Least-squares formulation of the model

The model presented here is directly derived from Ashok and Ben-Akiva (1993). We consider an analysis period divided into equal intervals $h = 1, \ldots, N$. The network is modeled by a directed graph $(\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{L}$ is the set of links. Origin-Destination (OD) pairs form a subset of $\mathcal{N} \times \mathcal{N}$ of cardinality n_{OD}. We denote by $x_h \in \mathbb{R}^{n_{OD}}$ the actual OD table capturing all trips departing during time interval h, and by x_h^H the associated historical OD table. The vector of deviations is denoted by $\partial x_h = x_h - x_h^H$. We assume that n_ℓ links from $\mathcal{L}$ are equipped with sensors able to count the number of vehicles during a given time interval. We note $y_{\ell h}$ the number of vehicles crossing sensor ℓ during time interval h, and $y_h \in \mathbb{R}^{n_\ell}$ the vector gathering all such counts. The model is composed of the *transition equations*, capturing the dynamic of the system, and the *measurement equation*, mapping the state variables onto the data.

The transition equations are based on an auto-regressive process on the OD flow deviations, which provides a preliminary estimate of the OD flow. They are given by:

$$\partial x_h = \sum_{p=h-q'}^{h-1} f_h^p \partial x_p + w_h, \tag{3.1}$$

where f_h^p, a $n_{OD} \times n_{OD}$ matrix, represents the contribution of ∂x_p to ∂x_h, q' is the number of former time intervals influencing ∂x_h and w_h is a vector of random variables capturing the error. Note that f_h^p are usually sparse in most practical applications. Namely, f_h^p is often computed from linear regression models for each OD pair. In that case, f_h^p matrices are diagonal, as correlation across ODs are therefore ignored. We make the following assumptions on w_h:

- $E[w_h] = 0$,
- $E[w_h w_t'] = Q_h \delta_{ht}$, where Q_h is a $(n_{OD} \times n_{OD})$ variance-covariance matrix, and δ_{ht} is the Kronecker symbol.

The measurement equations capture the relationship between the state vari-

ables (OD deviations), and the measurements (sensor data):

$$y_h = \sum_{p=h-p'}^{h} a_h^p x_p + \nu_h, \tag{3.2}$$

where $y_h \in \mathbb{R}^{n_\ell}$ contains the sensor data for time interval h, a_h^p is a $n_\ell \times n_{OD}$ matrix, called the assignment matrix, mapping OD flows departing during interval p to link flows observed during interval h. It captures network topology, route choice assumptions and travel time. These matrices are usually sparse, as it is not common that all OD flows use all sensors on the network, at every departure time interval. Finally, p' is the maximum number of time intervals needed to travel between any OD pair and ν_h is a vector of random variables capturing the error measurement on sensor data during time interval h. We make the following assumptions on ν_h:

- $E[\nu_h] = 0$
- $E[\nu_h \nu_t'] = R_h \delta_{ht}$, where R_h is an $n_\ell \times n_\ell$ variance-covariance matrix.

The assignment matrices a_h^p capture three aspects of the transportation system: the network topology (link-path incidence), the route choice model and the travel time across the network. Clearly, in congested networks, the matrices depend on the prevailing traffic conditions. Unfortunately, incorporating that dependence into the model significantly complicates the OD estimation problem. Therefore, most approaches assume that they are given for the OD estimation, and iterate between OD estimation and traffic assignment until some sort of convergence is reached. It is namely the approach suggested by the DynaMIT system (Ben-Akiva et al., 2001). Recent techniques, based on bilevel optimization problems, include explicitly traffic equilibrium conditions in the model (Florian and Chen, 1993, Barceló and Casas, 1999). Such issues are out of the scope of this research, which focused on algorithmic enhancements.

Equation (3.2) is not based on deviations. Therefore, we prefer the following equivalent formulation:

$$\partial y_h = \sum_{p=h-p'}^{h} a_h^p \partial x_p + \nu_h, \tag{3.3}$$

where $\partial y_h = y_h - \sum_{p=h-p'}^{h} a_h^p x_p^H$.

We present now the least-squares formulation of the real-time dynamic OD estimation and prediction problem. The size of the problem depends on data availability. We assume that sensor data is available for time intervals 1 to k. The least-squares formulation is given by

$$\min_X \sum_{h=1}^{k} \|\Omega_h^{-1} C_h^N X - \Omega_h^{-1} z_h\|_2^2 + \sum_{h=k+1}^{N} \|\Omega_h^{-1} C_h^N X\|_2^2, \tag{3.4}$$

where

$$X = \begin{pmatrix} \partial x_1 \\ \vdots \\ \partial x_{N-1} \\ \partial x_N \end{pmatrix}, \tag{3.5}$$

and

$$z_h = \begin{pmatrix} 0_{n_{OD}\times 1} \\ \partial y_h \end{pmatrix}, \tag{3.6}$$

$$\Theta_h = \Omega_h \Omega_h^T = \begin{pmatrix} Q_h & 0 \\ 0 & R_h \end{pmatrix} = \begin{pmatrix} P_h P_h^T & 0 \\ 0 & S_h S_h^T \end{pmatrix}, \tag{3.7}$$

and

$$\begin{aligned} C_h^N &= \begin{pmatrix} 0 & \cdots & 0 & -f_h^{h-q'} & \cdots & \cdots & \cdots & -f_h^{h-1} & I & 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \cdots & 0 & a_h^{h-p'} & \cdots & a_h^{h-1} & a_h^h & 0 & \cdots & 0 \end{pmatrix} \\ &= \begin{pmatrix} C_h^u \\ C_h^d \end{pmatrix} \end{aligned} \tag{3.8}$$

if $q' > p'$, and

$$\begin{aligned} C_h^N &= \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 & -f_h^{h-q'} & \cdots & -f_h^{h-1} & I & 0 & \cdots & 0 \\ 0 & \cdots & 0 & a_h^{h-p'} & \cdots & \cdots & \cdots & a_h^{h-1} & a_h^h & 0 & \cdots & 0 \end{pmatrix} \\ &= \begin{pmatrix} C_h^u \\ C_h^d \end{pmatrix} \end{aligned} \tag{3.9}$$

if $q' \leq p'$. Note that negative values of $h - p'$ and $h - q'$ are meaningless, and associated matrices are just ignored in the formulation. In general, we will denote by C_h^m, $k \leq m \leq N$, the $n_{OD} + n_\ell \times m n_{OD}$ matrix obtained from C_h^N by dropping the appropriate number of zeros on the right.

This (huge) least-squares problem has $N n_{OD}$ unknowns and $N n_{OD} + k n_\ell$ equations. It captures both estimation and prediction of the OD tables. Indeed, for any time interval h within the horizon, the solution of (3.4) provides an estimation of the OD tables up to interval k, and a prediction of OD tables for intervals $k+1$ to N.

It is important to note here that matrices f_h^p in (3.1) and a_h^p in (3.3) are very sparse for most realistic problems. The solution algorithms must exploit this sparsity in order to be able to handle large-scale problems.

From a practical viewpoint, problem (3.4) may be intractable when the number of state variables $n_{OD}N$ is large. However, the estimation and prediction problems can be treated separately. For the OD estimation problem, the structure of matrices C_h^N defined by (3.8) and (3.9) is such that only $n_{OD}s'$ state variables are actually updated for the OD estimation at each time interval, where $s' = \max(p', q')$. This must obviously be exploited in the implementation of any algorithm. Once the estimated OD tables are available, the

predicted OD tables are obtained by a direct application of the auto-regressive process.

If $n_{OD}s'$ is still too large for a specific algorithm, the problem size must be reduced even more. This is achieved by keeping the state variables $\partial x_{k-s'}$, ..., $\partial x_{k-\tau-1}$ constant, and updating only $\partial x_{k-\tau}$, ..., ∂x_k, for a given τ such that $0 \leq \tau \leq s'$. This procedure has been adopted for the OD estimation and prediction model implemented in DynaMIT (Antoniou et al., 1997, Ben-Akiva et al., 2002, Ben-Akiva et al., 2001), with $\tau = 0$. Note that the procedure does not bias the results if all vehicles are observed during one of the time intervals k, $k-1$, ... $k-\tau$. It means that the sensors must be sufficiently close to each origin in the network, so that each vehicle can be observed during the first τ time intervals of its trip.

3.3 Solution algorithms

We present here two solution algorithms. The Kalman filter algorithm (Kalman, 1960) is designed to update the solution of a least-squares problem in a real-time context, as more data is made available. We show that applying the Kalman filter algorithm to our least-squares formulation leads to the exact same algorithm as Ashok and Ben-Akiva (1993). Then, we consider the LSQR algorithm, proposed by Paige and Saunders (1982), in order to exploit (i) the sparsity of the problem and (ii) the a priori solution as provided by the auto-regressive process.

3.3.1 Kalman filter

The Kalman filter algorithm solves (3.4) in an iterative way. The algorithm for a general incremental least-squares problem is described by Bertsekas (1995). We assume that the problem has been solved up to time interval $k-1$, with solution X_{k-1} and variance-covariance matrix H_{k-1}. The update of these quantities is made through a two stage process. The first stage incorporates the transition equation to obtain $\widehat{X}_k$ and $\widehat{H}_k$, while the second incorporates the measurement equation to obtain X_k and H_k. However, in order to obtain an efficient formulation, the special structure of the problem must be exploited, as described below. Incorporating the transition equation is equivalent to solve the following problem:

$$\min_X \left\| \left(\begin{array}{c|c} P_k^{-1} & 0 \\ \hline 0 & (\Omega_{k-1}^{tot})^{-1} \end{array} \right) \left(\begin{array}{c|c} -F_{k-1} & I \\ \hline C_{k-1}^{tot} & 0 \end{array} \right) X - \left(\begin{array}{c} 0 \\ \hline (\Omega_{k-1}^{tot})^{-1} z_{k-1}^{tot} \end{array} \right) \right\|^2 \quad (3.10)$$

where

$$F_{k-1} = \left(0 \cdots 0 \;\; f_k^{k-q'} \cdots f_k^{k-1} \right) \in \mathbb{R}^{n_{OD} \times (k-1) n_{OD}}, \quad (3.11)$$

and

$$C_{k-1}^{tot} = \begin{pmatrix} C_1^{k-1} \\ \vdots \\ C_{k-2}^{k-1} \\ C_{k-1}^{k-1} \end{pmatrix}, z_{k-1}^{tot} = \begin{pmatrix} z_1 \\ \vdots \\ z_{k-2} \\ z_{k-1} \end{pmatrix}, \Omega_{k-1}^{tot} = \begin{pmatrix} \Omega_1 & & 0 \\ & \ddots & \\ 0 & & \Omega_{k-1} \end{pmatrix}. \quad (3.12)$$

The dimensions of these matrices are reported in Table 3.1:

Eq.	Matrix	Rows	Columns
(3.10)	P_k	n_{OD}	n_{OD}
(3.10)	Ω_{k-1}^{tot}	$n_{OD} + n_\ell$	$n_{OD} + n_\ell$
(3.10)	F_{k-1}	n_{OD}	$(k-1)n_{OD}$
(3.10)	C_{k-1}^{tot}	$(k-1)(n_{OD} + n_\ell)$	$(k-1)n_{OD}$
(3.10)	X_{k-1}	$(k-1)n_{OD}$	1
(3.10)	z_{k-1}^{tot}	$(k-1)(n_{OD} + n_\ell)$	1
(3.13)	$\widehat{X}_k$	kn_{OD}	1
(3.14)	H_{k-1}	$(k-1)n_{OD}$	$(k-1)n_{OD}$
(3.14)	$\widehat{H}_k$	kn_{OD}	kn_{OD}
(3.23)	K_k	kn_{OD}	n_ℓ
(3.23)	R_k	n_ℓ	n_ℓ

Table 3.1: Matrix dimensions

Note that the lower part of (3.10) gathers the $k-1$ first terms of (3.4), and that the terms corresponding to the prediction problem have been dropped.

Lemma 1 *The solution of (3.10) is given by:*

$$\widehat{X}_k = \begin{pmatrix} I \\ F_{k-1} \end{pmatrix} X_{k-1} \quad (3.13)$$

with variance-covariance matrix

$$\widehat{H}_k = \left(\begin{array}{c|c} H_{k-1} & H_{k-1} F_{k-1}^T \\ \hline F_{k-1} H_{k-1} & F_{k-1} H_{k-1} F_{k-1}^T + Q_k \end{array} \right). \quad (3.14)$$

Proof. We present here how to deduce (3.13) and (3.14) from (3.10) using the normal equations.

Remember that the solution of a least-squares problem

$$\min_x \|Ax - b\|^2 \quad (3.15)$$

is obtained from the normal equation $A^T Ax = A^T b$. The solution is

$$(A^T A)^{-1} A^T b, \quad (3.16)$$

with variance-covariance

$$(A^T A)^{-1}. \tag{3.17}$$

The normal equation for (3.10) is

$$\begin{aligned}\left(\begin{array}{c|c} -F_{k-1}^T & (C_{k-1}^{tot})^T \\ \hline I & 0 \end{array}\right)\left(\begin{array}{c|c} Q_k^{-1} & 0 \\ \hline 0 & \Theta_{k-1}^{-1} \end{array}\right)\left(\begin{array}{c|c} -F_{k-1} & I \\ \hline (C_{k-1}^{tot}) & 0 \end{array}\right) X = \\ \left(\begin{array}{c|c} -F_{k-1}^T & (C_{k-1}^{tot})^T \\ \hline I & 0 \end{array}\right)\left(\begin{array}{c|c} Q_k^{-1} & 0 \\ \hline 0 & \Theta_{k-1}^{-1} \end{array}\right)\begin{pmatrix} 0 \\ z_{k-1}^{tot} \end{pmatrix}.\end{aligned} \tag{3.18}$$

From (3.17), we have that

$$\widehat{H}_k^{-1} = \left(\begin{array}{c|c} -F_{k-1}^T & (C_{k-1}^{tot})^T \\ \hline I & 0 \end{array}\right)\left(\begin{array}{c|c} Q_k^{-1} & 0 \\ \hline 0 & \Theta_{k-1}^{-1} \end{array}\right)\left(\begin{array}{c|c} -F_{k-1} & I \\ \hline (C_{k-1}^{tot}) & 0 \end{array}\right). \tag{3.19}$$

A direct multiplication of (3.14) and (3.19) shows that $\widehat{H}_k \widehat{H}_k^{-1} = I$. The solution (3.13) is obtained from (3.16) and (3.14). We have

$$\begin{aligned}&\left(\begin{array}{c|c} H_{k-1} & H_{k-1}F_{k-1}^T \\ \hline F_{k-1}H_{k-1} & F_{k-1}H_{k-1}F_{k-1}^T + Q_k \end{array}\right)\left(\begin{array}{c|c} -F_{k-1}^T & (C_{k-1}^{tot})^T \\ \hline I & 0 \end{array}\right)\left(\begin{array}{c|c} Q_k^{-1} & 0 \\ \hline 0 & \Theta_{k-1}^{-1} \end{array}\right)\begin{pmatrix} 0 \\ z_{k-1}^{tot} \end{pmatrix} \\ = &\left(\begin{array}{c|c} -H_{k-1}F_{k-1}^T + H_{k-1}F_{k-1}^T & H_{k-1}(C_{k-1}^{tot})^T \\ \hline -F_{k-1}H_{k-1}F_{k-1}^T + F_{k-1}H_{k-1}F_{k-1}^T + Q_k & F_{k-1}H_{k-1}(C_{k-1}^{tot})^T \end{array}\right)\begin{pmatrix} 0 \\ \Theta_{k-1}^{-1} z_{k-1}^{tot} \end{pmatrix} \\ = &\left(\begin{array}{c|c} 0 & H_{k-1}(C_{k-1}^{tot})^T \\ \hline Q_k & F_{k-1}H_{k-1}(C_{k-1}^{tot})^T \end{array}\right)\begin{pmatrix} 0 \\ \Theta_{k-1}^{-1} z_{k-1}^{tot} \end{pmatrix} \\ = &\begin{pmatrix} H_{k-1}(C_{k-1}^{tot})^T \Theta_{k-1}^{-1} z_{k-1}^{tot} \\ F_{k-1}H_{k-1}(C_{k-1}^{tot})^T \Theta_{k-1}^{-1} z_{k-1}^{tot} \end{pmatrix} = \begin{pmatrix} I \\ F_{k-1} \end{pmatrix} H_{k-1}(C_{k-1}^{tot})^T \Theta_{k-1}^{-1} z_{k-1}^{tot}.\end{aligned}$$

We finally obtain (3.13) by noting that

$$X_{k-1} = H_{k-1}(C_{k-1}^{tot})^T \Theta_{k-1}^{-1} z_{k-1}^{tot}.$$

□

The measurement equation is incorporated now as follows, again based on Bertsekas (1995).

$$H_k = \widehat{H}_k + (C_k^d)^T R_k^{-1} C_k^d, \tag{3.20}$$

$$X_k = \widehat{X}_k + H_k^{-1}(C_k^d)^T (R_k^{-1} \partial y_k - R_k^{-1} C_k^d \widehat{X}_k) \tag{3.21}$$

From the Sherman-Morrison-Woodbury formula (Golub and Van Loan, 1996), we have that

$$H_k^{-1} = \left(I - K_k C_k^d\right) \widehat{H}_k^{-1}, \tag{3.22}$$

where

$$K_k = \widehat{H}_k^{-1}(C_k^d)^T \left(R_k + C_k^d \widehat{H}_k^{-1}(C_k^d)^T\right)^{-1}. \tag{3.23}$$

Lemma 2

$$X_k = \widehat{X}_k + K_k(\partial y_k - C_k^d \widehat{X}_k). \tag{3.24}$$

Proof. Using (3.22) in (3.21) we obtain

$$X_k = \widehat{X}_k + \left(I - K_k C_k^d\right) \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} \left(\partial y_k - C_k^d \widehat{X}_k\right). \tag{3.25}$$

Denoting $g_k = \left(\partial y_k - C_k^d \widehat{X}_k\right)$, we show that

$$\begin{aligned} \left(I - K_k C_k^d\right) \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k &= \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k - K_k C_k^d \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k \\ &= K_k g_k. \end{aligned} \tag{3.26}$$

Indeed

$$\begin{aligned} & K_k C_k^d \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k \\ = \; & \widehat{H}_k^{-1} (C_k^d)^T \left(R_k + C_k^d \widehat{H}_k^{-1} (C_k^d)^T\right)^{-1} C_k^d \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k \\ = \; & \widehat{H}_k^{-1} (C_k^d)^T \left(R_k + C_k^d \widehat{H}_k^{-1} (C_k^d)^T\right)^{-1} \left(C_k^d \widehat{H}_k^{-1} (C_k^d)^T + R_k\right) R_k^{-1} g_k \\ & - \widehat{H}_k^{-1} (C_k^d)^T \left(R_k + C_k^d \widehat{H}_k^{-1} (C_k^d)^T\right)^{-1} R_k R_k^{-1} g_k \\ = \; & \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k - \widehat{H}_k^{-1} (C_k^d)^T \left(R_k + C_k^d \widehat{H}_k^{-1} (C_k^d)^T\right)^{-1} g_k \\ = \; & \widehat{H}_k^{-1} (C_k^d)^T R_k^{-1} g_k - K_k g_k, \quad \text{from (3.23).} \end{aligned}$$

□

Note that equations (3.13), (3.14), (3.22), (3.23) and (3.24) are equivalent to the algorithm proposed by Ashok and Ben-Akiva (1993). This result is important, as it proves that our approach of the problem is actually equivalent to theirs.

3.3.2 LSQR

LSQR is an iterative method for solving the least-squares problem

$$\min_{x \in \mathbb{R}^n} \|Ax - b\|_2^2 \tag{3.27}$$

when A is large and sparse. Proposed by Paige and Saunders (1982), it is analytically equivalent to the conjugate gradient method, which is iterative by nature. Its convergence is theoretically achieved within at most n iterations. LSQR, based on two bi-diagonalization procedures, generates a sequence of x_k such that the associated sequence of residual's norms monotonically decreases. It exhibits better numerical properties than the conjugate gradient method, especially when A is ill-conditioned. A key property of this algorithm is that the matrix A is used only to compute products of the form Ax or $A^T y$, where x and y are vectors of appropriate dimensions, which is particularly attractive for large sparse problems. Indeed, A does not need to be explicitly constructed

and stored, which is a particularly appealing feature for solving (3.4), given its specific structure.

LSQR is detailed by Paige and Saunders (1982). In our context this algorithm has two drawbacks. First it is not designed for real-time applications. Second it is designed to start from 0. However, it can be adapted to solve real-time problems. First, its iterative nature allows for an easy update of a previous estimate $\bar{x}$. Defining $y = x - \bar{x}$, (3.27) can be written as

$$\min_{y \in \mathbb{R}^n} \|Ay - (b - A\bar{x})\|_2^2. \tag{3.28}$$

We denote by

$$x^* = \mathrm{LSQR}(A, b, \bar{x}) = \bar{x} + \mathrm{argmin}_{y \in \mathbb{R}^n} \|Ay - (b - A\bar{x})\|_2^2. \tag{3.29}$$

Second, it can be applied in a real-time context as follows.

Initialize When no sensor data is available, historical OD tables are the best estimates. Therefore, we set $X_0 = 0$, that is $\partial x_h = 0$, $h = 1, \ldots, N$ and $k = 0$.

For $k = 1, \ldots, N$ At each interval k, we incorporate more sensor data, and update the estimated and predicted OD tables accordingly as follows

$$X_k = \mathrm{LSQR}\left(\sum_{h=1}^{k} \Omega_h^{-1} C_h^k, \sum_{h=1}^{k} \Omega_h^{-1} z_h, X_{k-1} \right). \tag{3.30}$$

Contrarily to LSQR, the Kalman filter algorithm is incremental by nature. At each time interval k, it involves only the matrices C_k and Ω_k, and the vector z_k, while LSQR involves matrices from all previous time intervals as well (see (3.30)). Consequently, the size of the problem grows with time, and LSQR does not look like an appealing candidate for real-time applications at first glance. This is probably one of the reasons why the Kalman filter algorithm has been widely proposed for real-time applications in the literature. On the other hand, the Kalman filter ignores and destroys the sparsity of the matrices (see (3.13), (3.14), (3.22), (3.23) and (3.24)). In order for LSQR to be applied in a real-time context, the number of terms in (3.30) must be kept constant. Therefore, we propose to replace (3.30) by

$$X_k = \mathrm{LSQR}\left(\sum_{h=k-r'}^{k} \Omega_h^{-1} C_h^k, \sum_{h=k-r'}^{k} \Omega_h^{-1} z_h, X_{k-1} \right), \tag{3.31}$$

where r' must be greater or equal to s'. The choice of r' is a trade-off between accuracy of the solution, and computation burden. Indeed, ignoring the terms corresponding to time intervals 1 to $k - r' - 1$ slightly biases the solution. Actually, it is equivalent to ignore the estimation error of those time intervals, by not propagating the variance-covariance matrix.

Note that the bias can be reduced, while keeping the problem's sparsity, by propagating only the variance of the estimators. This would keep the variance-covariance matrix diagonal. However, we do not investigate this possibility. Indeed, it appears from the experiments we have conducted (see Section 3.5) that the bias associated with (3.31) is not significant. Finally, the OD prediction problem is not directly solved by LSQR, as it simply amounts to applying the auto-regressive process to the estimated deviations.

3.4 Theoretical comparisons

We compare here the numerical complexity of both algorithms. Following Golub and Van Loan (1996), we count the number of floating point operations (flops) associated with each algorithm. Note that if A is an $m \times n$ matrix, and B is an $n \times p$ matrix, the product AB takes $2mnp$ flops. If C is an invertible matrix of dimension n, computing its inverse takes $2n^3$ flops.

An upper bound on the total number of flops to perform p iterations of the LSQR algorithm is

$$2Cu + 6\overline{m} + 4\overline{n} + p(2Cu + 6\overline{m} + 10\overline{n} + 25), \tag{3.32}$$

where Cu is the number of flops required to compute the matrix-vector products $\Omega_k^{-1}C_kX$ and $(\Omega_k^{-1}C_k)^TX$, $\overline{m} = (r'+1)(n_{OD}+n_\ell)$ and $\overline{n} = (s'+1+r')n_{OD}$.

We denote by d_A the density of a sparse matrix A, that is the number of nonzero entries divided by the total number of entries. We have that

$$d_{C_k} = \frac{(r'+1)n_{OD} + q'(r'+1)n_{OD}^2 d_f + (p'+1)(r'+1)n_\ell n_{OD} d_a}{(s'+1+r')n_{OD}(r'+1)(n_{OD}+n_\ell)}, \tag{3.33}$$

and

$$d_{\Omega_k^{-1}} = \frac{(r'+1)(n_{OD}^2 d_{Q^{-1}} + n_\ell^2 d_{R^{-1}})}{(r'+1)^2(n_{OD}+n_\ell)^2}, \tag{3.34}$$

where d_f is an upper bound on the density of matrices f_h^p defined in (3.1), and d_a is an upper bound on the density of assignment matrices a_h^p defined in (3.2). $d_{Q^{-1}}$ and $d_{R^{-1}}$ are similarly defined. Consequently,

$$\begin{aligned} Cu &= 2(s'+1+r')(r'+1)(n_\ell+n_{OD})n_{OD}d_{C_k} \\ &\quad +2(r'+1)^2(n_{OD}+n_\ell)^2 d_{\Omega_k^{-1}}. \end{aligned} \tag{3.35}$$

For the Kalman filter algorithm, we assume that the result of the product of two sparse matrices is dense to obtain the number of flops for each equation.

Eq. (3.13) $n_{OD} + q'd_f n_{OD}^2$,

Eq. (3.14) $2(q')^3 n_{OD}^3 d_f + n_{OD}^2 d_Q$,

Eq. (3.22) $(s'+1)n_{OD}(2n_\ell d_a + 2(s'+1)^2 n_{OD}^2 + 1)$,

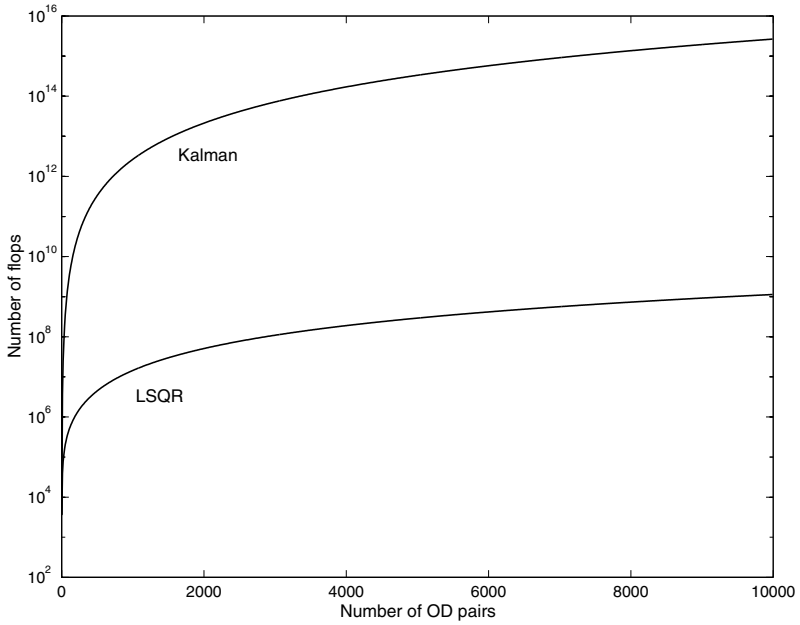


Figure 3.1: Kalman-LSQR flops comparison

Eq. (3.23) $4n_\ell(s'+1)^2n_{OD}^2d_a + n_\ell^2 d_R + 2n_\ell^{3'} + 2(s'+1)n_{OD}n_\ell^2$,

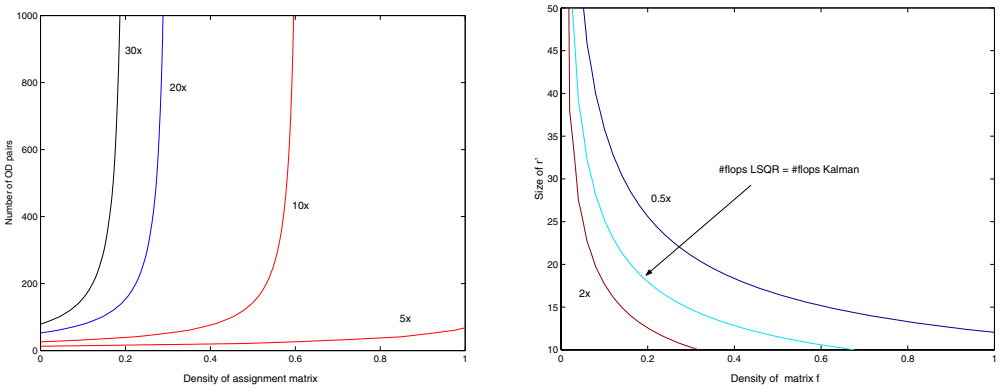
Eq. (3.24) $2(s'+1)n_{OD}(1+(s'+1)n_{OD}d_a)$.

We illustrate these formulas in Figure 3.1, where the (logarithm of the) number of flops for each algorithm is plotted as a function of the number of ODs. We assume that $n_\ell = n_{OD}/10$, the variance-covariance matrices and the transition matrix are diagonal, and that $p' = r' = 10$ and $q' = 9$. The density of the assignment matrix is $d_a = 5\%$.

In Figure 3.2, we analyze scenarios where the relative performance of both algorithms is given. In Figure 3.2(a), we assume that $n_\ell = n_{OD}/10$, the variance-covariance matrices and the transition matrix are diagonal, and that $p' = r' = 10$ and $q' = 9$. The plotted curves represent combinations of values for n_{OD} and d_a such that the LSQR algorithm is 5, 10, 20 and 30 times (resp.) faster than the Kalman filter algorithm. In Figure 3.2(b), we have $n_{OD} = 1000$, $n_\ell = 100$ and $d_a = 2\%$. The plotted curves represent combinations of values for r' and d_f such that the LSQR algorithm performs as well, twice faster ($2\times$) and twice slower ($0.5\times$) (resp.) than the Kalman filter algorithm.

It clearly appears from these plots that LSQR is much more efficient than Kalman filter when sparse matrices are involved, which often occurs in practice. Interestingly, when the matrix sparsity is important (low values of d_f is Figure 3.2(b)), the r' parameter introduced to simplify the model in (3.31) can be set to a high value, while keeping the performance gain significant.

We complete the theoretical analysis by a simplification of the flops counting formulas, in order to obtain a level of magnitude. For this purpose, we assume



(a) n_{OD} vs. d_a (b) r' vs. d_f

Figure 3.2: Kalman-LSQR flops equivalence

that $n_{OD} = n^2$ and $n_\ell = \delta n$, where n is the number of nodes in the network, and δ is the average degree of the nodes. It is pessimistic, as all nodes are supposed to be both origins and destinations, and all links are assumed to be equipped with sensors.

If n is large, the dominant term is

$$4(p+1)(r'+1)n^4(q'd_f + d_{Q^{-1}}). \tag{3.36}$$

The sparsity of the auto-regressive process is therefore critical for the performance of the LSQR algorithm. A similar analysis for the Kalman filter algorithm leads to the following dominant term:

$$2(s'+1)^3 n^6(2d_f+1). \tag{3.37}$$

We deduce from (3.36) and (3.37) that if the number of LSQR iterations p is such that

$$p \leq \frac{(s'+1)^3(2d_f+1)}{2(r'+1)(q'd_f + d_{Q^{-1}})} n^2, \tag{3.38}$$

than LSQR is more efficient than Kalman filter. In a real-time context, the number of LSQR iterations are usually low as the starting point is close to the solution when traffic conditions are more or less stable. In the worst case, LSQR theoretically performs $(r'+1)n^2$ iterations. In order for LSQR to be better, the density of the transition matrices has to be such that

$$d_f \leq \frac{(s'+1)^3 - (r'+1)^2 d_{Q^{-1}}}{(r'+1)^2 q' - 2(s'+1)^3}. \tag{3.39}$$

Again, we observe that high values of r' are acceptable if the density of the transition matrices is low.

When both the transition matrices f_h^p and associated variance-covariance matrices are diagonal, the density of the assignment matrix becomes the dominant parameter in the flops computation. Indeed, assuming that $d_f = 1/n^2$ and $d_{Q^{-1}} = 1/n^2$, the dominant term for Cu in the number of flops (3.32) is

$$4(p+1)(r'+1)p'\delta n^3 d_a. \tag{3.40}$$

In that case, the complexity of LSQR depends on the density of the assignment matrix, and not any more on the density of the transition equations.

Considering again the worst case where LSQR performs $(r'+1)n^2$ iterations, the following condition must be verified in order for LSQR to be more efficient that the Kalman filter algorithm.

$$d_a \leq \frac{(s'+1)^3}{2(r'+1)^2 p'\delta} n. \tag{3.41}$$

Note that for large values of n and reasonable values of r', the density of the assignment matrix is irrelevant, and LSQR is systematically better than Kalman filter. This is illustrated by Figure 3.2(a).

3.5 Numerical comparisons

We provide now an empirical comparison of the Kalman filter and LSQR algorithms to solve (3.4). Both algorithms have been implemented in Matlab (The Mathworks Inc., 1994), using the sparse matrices structure. The implementation of the Kalman filter algorithm that we use for these numerical comparisons actually uses less number of flops than the theoretical number estimated in Section 3.4. First, we use synthetic data in order to illusrate the accuracy of both approaches, when the "true" OD tables are known by the analyst. The objective is to illustrate the impact on the limitation of the number of terms in (3.31). Then, we present two case studies to compare the computational performance of the algorithms: a medium-scale model for the Central Artery/Third Harbor Tunnel (CA/T) network in Boston (Ma), and a large-scale model in Irvine (Ca).

3.5.1 Synthetic data

We consider the network depicted in Figure 3.3, with three OD pairs {(1,5), (1,6), (2,6)}, and a time horizon of N = 15 time intervals of T minutes each. For the sake of simplicity, we assume for each link a travel time of T minutes and an infinite capacity. Consequently, $p' = 3$. We also assume that $q' = 2$ and, therefore $s' = 3$.

The "true" OD flows for pairs $(1,5)$ and $(1,6)$ are given in Table 3.2, where the unit is a number of cars per time interval (T minutes). The flows for OD pair $(2,6)$ are twice these values. Note that time intervals -4 to 0 are used to warm up the simulation and to avoid starting with an empty network.

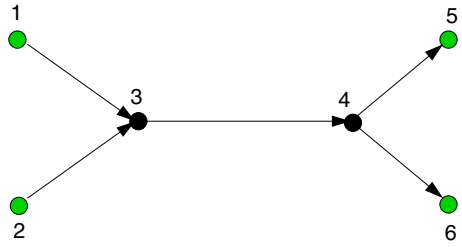


Figure 3.3: Simple network

Time int.	OD	Time int.	OD	Time int.	OD	Time int.	OD
-4	36	1	30	6	12	11	42
-3	12	2	48	7	12	12	24
-2	30	3	12	8	60	13	42
-1	12	4	18	9	42	14	12
0	36	5	36	10	66	15	18

Table 3.2: "True" demand for the simple network

The historical OD tables have been obtained by a random perturbation of the true OD. The link flows resulting from the assignment of the true OD tables have also been perturbed to obtain the sensor data. The auto-regressive process is such that $f_h^p = I$, for $p = h-3, h-2, h-1$. The variance-covariance matrices Q_k are diagonal, with the OD flows of the last time interval on the diagonal. Variance-covariance matrices R_k are diagonal with variance arbitrarily set to 1.

The relative error on estimated OD flows, that is

$$\frac{\|X_{true} - X_{estimated}\|}{\|X_{estimated}\|}, \tag{3.42}$$

obtained by each algorithm is illustrated by Figure 3.4. It appears, as expected, that the Kalman filter algorithm is more accurate than the LSQR algorithm from the 5th time interval on, as $r' + 1 = 4$. It is particularly noticeable that the difference remains almost constant (as it depends mainly on r') and is negligible.

3.5.2 Case-studies

DynaMIT is a state-of-the-art, real-time computer system for traffic estimation, prediction, and generation of traveler information and route guidance. It supports the operation of Advanced Traveler Information Systems (ATIS) and Advanced Traffic Management Systems (ATMS) at Traffic Management Centers. DynaMIT is the result of about 10 years of intense research and development at the Intelligent Transportation Systems Program of the Massachusetts

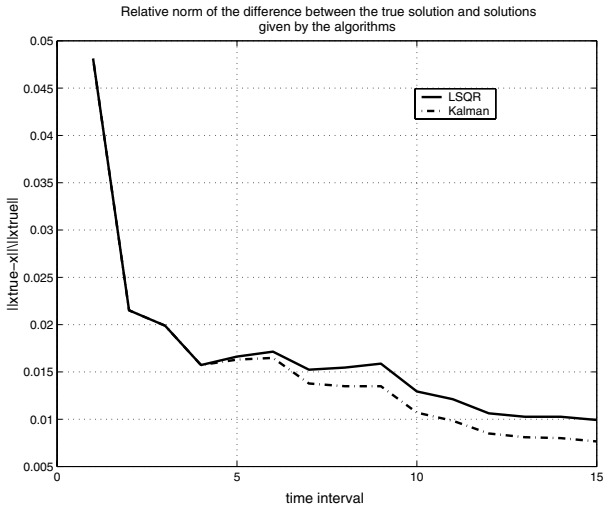


Figure 3.4: Error in the solution given by the algorithms

Institute of Technology (for description and details, see Ben-Akiva et al., 2002 and Bottom et al., 1999). DynaMIT's OD estimation and prediction algorithm (Antoniou et al., 1997) is a Kalman filter algorithm directly derived from Ashok and Ben-Akiva (1993). We have used DynaMIT to obtain assignment matrices for both case studies. We have also compared the results obtained with DynaMIT with those obtained with Matlab, in order to verify the algorithms implementation.

The first network is the Central Artery/Third Harbor Tunnel network, currently under construction (see Figure 3.5). It is a medium-scale network, with 211 links and 183 nodes. We consider a scenario with five origins and two destinations for a total of 10 OD pairs, and 35 link counts. We simulate 60 minutes during the morning from 7:00am to 8:00am. This simulation period is divided into 15 minutes time intervals. The results are described in Section 3.5.2.

The second network contains the major highways I-5, I-405 and CA-133 around Irvine, Ca. It contains also arterial roads in a triangular area defined by I-5, I-405 and Jeffrey road (see Figure 3.6). It is a large-scale network, with 618 links, 296 nodes and 627 OD pairs. We simulate 60 minutes during the morning from 7:15am to 8:15am. This simulation period is divided into 15 minutes time intervals. The results are described in Section 3.5.2. Irvine network data comes from a traffic management center in Irvine, California.

Central artery network

We solve the OD estimation problem for the CA/T network with $p' = 0$ and $q' = r' = 1$. In table 3.3, we report (i) the average number of flops per time interval for the Kalman filter algorithm, as reported by Matlab, (ii) the same

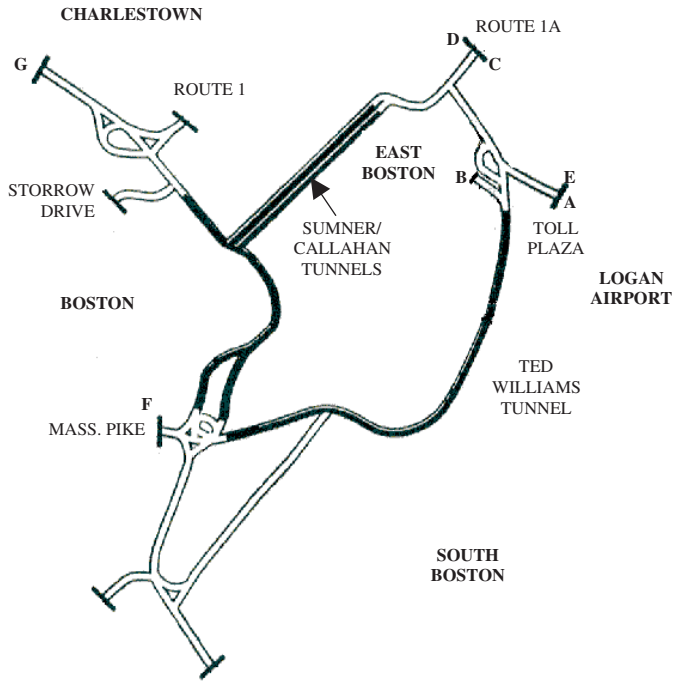


Figure 3.5: Central Artery/Third Harbor Tunnel network

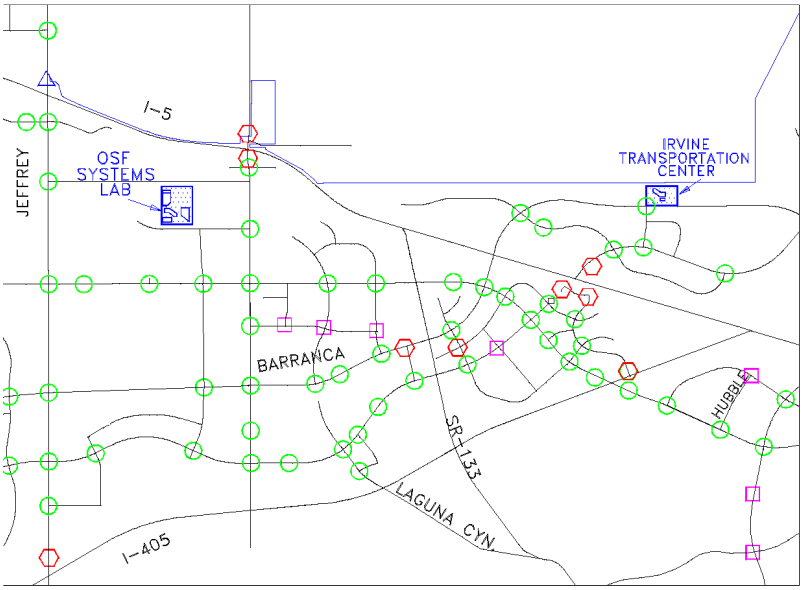


Figure 3.6: Irvine network

information for LSQR algorithm, (iii) the relative root mean squared error (RRMSE), that is

$$\frac{\sqrt{n_\ell \sum_{k=1}^{N} \sum_{j=1}^{n_{OD}} \left((\partial x_k)_j^{\text{Kalman}} - (\partial x_k)_j^{\text{LSQR}} \right)^2}}{\sum_{k=1}^{N} \sum_{j=1}^{n_{OD}} |((\partial x_k)_j^{\text{Kalman}}|} \tag{3.43}$$

and the relative mean error (RME), that is

$$\frac{\sum_{k=1}^{N} \sum_{j=1}^{n_{OD}} \left| (\partial x_k)_j^{\text{Kalman}} - (\partial x_k)_j^{\text{LSQR}} \right|}{\sum_{k=1}^{N} \sum_{j=1}^{n_{OD}} |((\partial x_k)_j^{\text{Kalman}}|}, \tag{3.44}$$

for various values of the ATOL parameter, ATOL being the tolerance on the normalized least-squares residual used as a stopping criterion for the LSQR algorithm (see Paige and Saunders, 1982).

ATOL	flops Kalman	flops LSQR	RRMSE	RME
10^{-2}	38800	4100	0.979	0.766
10^{-3}	38800	5753	0.076	0.054
10^{-4}	38800	5949	0.044	0.037
10^{-5}	38800	6471	0.044	0.034

Table 3.3: Comparison for the CA/T network

In appears clearly from Table 3.3 that the empirical results are consistent with the theoretical analysis, and that LSQR significantly outperforms the Kalman filter algorithm. This instance, where LSQR is 6 times better than Kalman in the worst case (ATOL=10^{-5}) is representative of other experiments on problems of similar characteristics. LSQR also allows a trade-off between the results accuracy and computational burden. This is often critical for real-time applications. Setting the ATOL parameter to 10^{-2} produces an algorithm almost 10 times faster than Kalman, with a reasonable reduction of accuracy.

As a final note, the theoretical number of flops for LSQR (see Section 3.4) is about 10000, and for Kalman is about 160000. The discrepancy between theoretical and actual numbers of flops is due to the simplifying assumptions used in Section 3.4.

Irvine network

We solve the OD estimation problem for the Irvine network with $p' = 0$ and $q' = r' = 1$. In table 3.4, we report (i) the average number of flops per time interval for the Kalman filter algorithm, as reported by Matlab, (ii) the same information for LSQR algorithm, (iii) the RRMSE (3.43) and (iv) the MSE (3.44).

ATOL	flops Kalman	flops LSQR	RRMSE	RME
10^{-2}	$7.38\ 10^7$	$5.42\ 10^5$	0.4262	0.3020
10^{-3}	$7.38\ 10^7$	$1.30\ 10^6$	0.3713	0.2380
10^{-4}	$7.38\ 10^7$	$2.34\ 10^6$	0.1435	0.1149
10^{-5}	$7.38\ 10^7$	$3.23\ 10^6$	0.1425	0.1146

Table 3.4: Comparison for the Irvine network

The LSQR algorithm solves the problem from 23 times (ATOL=10^{-5}) to 136 times (ATOL=10^{-2}) faster than Kalman. As predicted by the theoretical analysis, the advantage of using LSQR becomes more significant when the size of the problem increases.

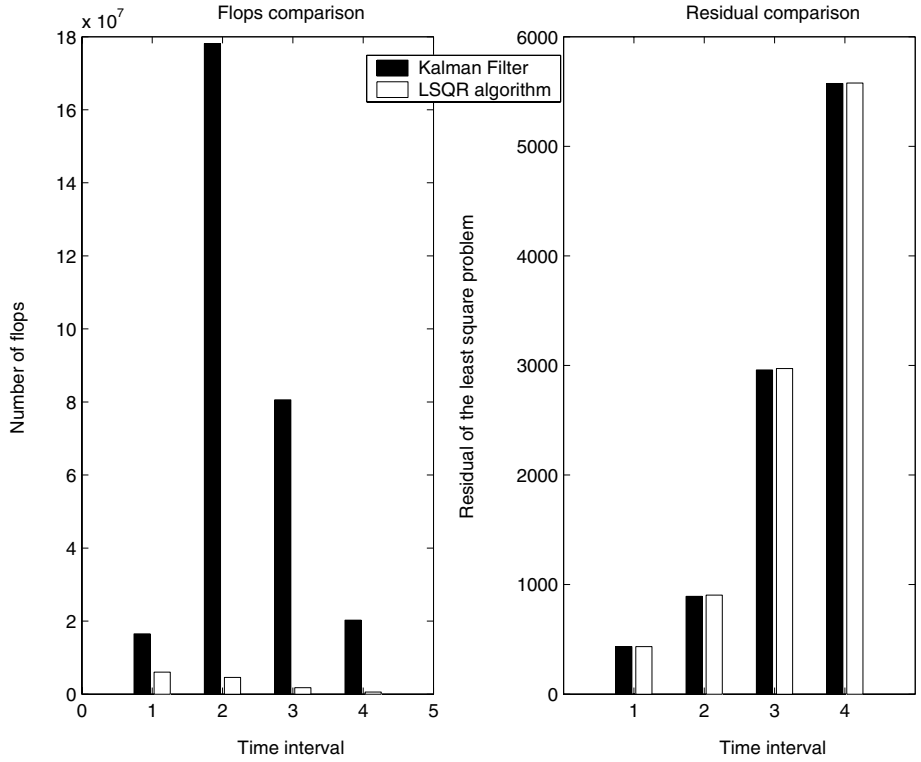


Figure 3.7: Results for the Irvine network

Figure 3.7 shows for each time interval and for each algorithm, the number of flops on the left and the value of the residual on the right, *i.e.* $\|\Omega_h^{-1}C_hX_h - \Omega_h^{-1}z_h\|$. It clearly demonstrates the superiority of LSQR method in large-scale case, with a similar quality of the result.

Note that the theoretical number of flops for LSQR is about $9\ 10^6$, and for

Kalman is 4 10^9. Again, the discrepancy between theoretical and actual number of flops in Table 3.4 is due to the simplifying assumptions used in Section 3.4.

Note that we prefer to mention flops instead of computation time to decrease the impact of a specific computer hardware, MATLAB's overhead, memory access and implementation. However, these times provide us with the same qualitative conclusions. As an example, Table 3.5 reports the run time of each algorithm on Irvine case study. These times have been obtained by running MATLAB on a Dell PowerEdge 6300 with 4 Intel Pentium II, 500 Mhz, 1 Gb memory.

Time Interval	1	2	3	4
Kalman Filter	1.03	5.69	2.35	1.27
LSQR algorithm	1.66	1.49	0.73	0.24

Table 3.5: CPU Time computation in seconds for the Irvine network

Finally, we have run the Irvine case study with $p' = r' = 4$ and $q' = 3$. In that case, we were not able to solve it with the Kalman filter algorithm, which exhausted the available memory in Matlab. The LSQR algorithm has been able to solve the problem in about 6.7 10^7 flops. Note that this is less than the number of flops reported for the Kalman filter algorithm in Table 3.4.

3.6 Conclusion

In this chapter, we have proposed a least-squares formulation of the real-time dynamic OD estimation and prediction problem, based on a combination of the approaches by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993). In order to emphasize the model's validity, we have shown that applying the Kalman filter algorithm as presented by Bertsekas (1995) leads to the exact same algorithm as Ashok and Ben-Akiva (1993).

The proposed formulation enables to directly use algorithm LSQR to solve the problem. Proposed by Paige and Saunders (1982), LSQR is a numerically robust conjugate gradient algorithm designed to solve large-scale sparse least-squares problems. Because it is not designed for real-time applications, we have imposed a simplification to maintain the size of the problem constant over time. This simplification amounts not to propagate the variance-covariance matrix of old estimated matrices.

Both the theoretical estimation of the flops and empirical comparisons on real data exhibit a significantly better performance for the LSQR algorithm, in the presence of sparse matrices. We have also shown that the model simplification has a limited impact on the quality of the solution.

The Kalman filter approach based on the normal equations cannot afford large-scale problems, as it involves the multiplication and inversion of very large matrices. The computational complexity of the LSQR algorithm is based only

on its ability to multiply a large matrix by a vector. If the large matrix is very sparse, as it is often the case in practice, such a procedure can be implemented efficiently. Also, the iterative nature of LSQR allows, contrarily to Kalman, to exploit previous estimates when the traffic conditions are stable, performing less iterations to converge to the solution.

A direct extension of the algorithm presented here is obtained when the OD deviations are constrained by lower and upper bounds. One important motivation is to avoid negative OD flows when the deviations are added to the historical values. The bound-constrained LSQR algorithm proposed by Bierlaire et al. (1991) can be considered in that case.

We emphasize that the least-squares formulation adopted is well adapted when additional data can be considered in order to improve the quality of the estimated OD. For example, license plate data collected in parking lot (Bierlaire and Toint, 1995) or probe vehicles data based on GPS, ETC or cellular phone technologies (Smith et al., 2001). Contrarily to the Kalman filter algorithm, which requires to re-derive the equations, the LSQR algorithm can be used as is, with the extended formulation.

Finally, the ideas proposed in this research are not restricted to the specific model formulation proposed by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993). Our decision to adopt this model is to provide a formal and unbiased comparison between the algorithms in a given context. But our ideas can be applied to any model that aims to solve a real-time problem with a Kalman filter approach.

Chapter 4

Generalized Secant Method

Contents

4.1 Introduction

We consider the standard problem of identifying the solution of a system of nonlinear equations

$$F(x) = 0 \tag{4.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable function. Since Newton, this problem has received a tremendous amount of attention. Newton's method and its many variations are still intensively analyzed and used in practice. The philosophy of Newton-like methods is to replace the nonlinear function F by a linear model, which approximates F in the neighborhood of the current iterate. The original Newton method invokes Taylor's theorem and uses the first derivative matrix (or Jacobian) to construct the linear model. When the Jacobian is too expensive to evaluate, secant methods build the linear model based on the secant equation. Because secant methods exhibit a decent rate of convergence under mild assumptions (q-super-linear), they have been intensively analyzed in the literature.

The secant equation imposes that the linear model perfectly matches the nonlinear function F at two successive iterates. If the number of unknowns n is strictly greater than 1, an infinite number of linear models verify the secant equation. Therefore, each secant method derives a specific update formula which arbitrarily picks one linear model among them. The most common strategies are called "least-change updates" and select the linear model that minimizes the difference between two successive models.

In this chapter, we provide a class of algorithms generalizing these methods. They are also based on a linear model. Instead of using only two successive iterates to determine it, we maintain a "population" of iterates. Indeed, in the presence of expensive function evaluation, we want to incorporate all the acquired information about the function to calibrate at best the model. Also, we expect a population-based approach to be more robust in the presence of noise in the function.

An important feature of our method is that we do not impose an exact match between the model and the function. Instead, we use a least-squares approach to request that the model matches the function "as much as possible". This class of algorithms exhibits a faster convergence and a greater robustness than quasi-Newton methods for most numerical tests that we have performed (Section 4.5) at a cost of substantial linear algebra computation. Therefore it is valuable when the cost of evaluating F is high in comparison with the numerical algebra overhead. In this chapter, we present the class of algorithms (Section 4.3) and prove that they are locally convergent (Section 4.4).

4.2 Quasi-Newton methods

Quasi-Newton methods consider at each iteration the linear model

$$L_k(x; B_k) = F(x_k) + B_k(x - x_k) \tag{4.2}$$

that approximates $F(x)$ in the neighborhood of x_k and computes x_{k+1} as a solution of the linear system $L_k(x; B_k) = 0$. Consistently with most of the publications on this topic, quasi-Newton methods can be summarized as methods based on the following iterations:

$$x_{k+1} = x_k - B_k^{-1} F(x_k). \tag{4.3}$$

followed by the computation of B_{k+1}. Of course the most illustrious quasi-Newton method is the pure Newton method where $B_k = J(x_k) = \nabla F(x_k)^T$ is the Jacobian of F evaluated at x_k, that is a $n \times n$ matrix such that entry (i, j) is $\partial F_i / \partial x_j$. We refer the reader to Dennis and Schnabel (1996) for an extensive analysis of Newton and quasi-Newton methods.

Broyden (1965) proposes a class of quasi-Newton methods based on the *secant equations*, imposing the linear model L_{k+1} to exactly match the nonlinear function at iterates x_k and x_{k+1}, that is

$$\begin{array}{ll} L_{k+1}(x_k) & = F(x_k), \\ L_{k+1}(x_{k+1}) & = F(x_{k+1}). \end{array} \tag{4.4}$$

Subtracting these two equations and defining $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$ we obtain the classical secant equation:

$$B_{k+1} s_k = y_k. \tag{4.5}$$

Clearly, if the dimension n is strictly greater than 1, there is an infinite number of matrices B_{k+1} satisfying (4.5). An arbitrary decision must consequently be made. The "least-change secant update" strategy, proposed by Broyden (1965), consists in selecting among the matrices verifying (4.5) the one minimizing variations between two successive models $L_k(x)$ and $L_{k+1}(x)$. It leads to the following update formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)\, s_k^T}{s_k^T s_k}. \tag{4.6}$$

This method has been very successful, and has been widely adopted in the field. However, we believe that the philosophy of interpolating the linear model on only two iterates and so forgetting all the information given by previous iterates could be too restrictive, incurring the risk that the arbitrariness introduced by the method plays an overly important role. Therefore, we propose here to use more than two past iterates to build the linear model, expecting a better approximation of the actual tangent model.

This idea has already been considered. Dennis and Schnabel (1996) say that "In fact, multivariable generalizations of the secant method have been proposed ... but none of them seem robust enough for general use." Later, they write "Perhaps the most obvious strategy is to require the model to interpolate $F(x)$ at other past points... One problem is that the directions tend to be linearly dependent or close to it, making the computation of (the approximation matrix) a poorly posed numerical problem".

There are few attempts to generalize this approach in the literature. A first generalization of the secant method is the *sequential secant method* proposed by Wolfe (1959) and discussed by Ortega and Rheinboldt (1970). The idea is to impose exact interpolation of the linear model on $n+1$ iterates instead of 2:

$$L_{k+1}(x_{k-j}) = F(x_{k-j}), \quad j = 0, 1, \ldots, n. \tag{4.7}$$

or, equivalently,

$$B_{k+1} s_{k-j} = y_{k-j}, \quad j = 0, \ldots, n, \tag{4.8}$$

where $s_i = x_{k+1} - x_i$, and $y_i = F(x_{k+1}) - F(x_i)$, for all i. If the vectors $s_k, s_{k-1}, \ldots, s_{k-n+1}$ are linearly independent, there exists exactly one matrix B_{k+1} satisfying (4.8), which is

$$B_{k+1} = \tilde{Y}_{k+1} \tilde{S}_{k+1}^{-1} \tag{4.9}$$

where $\tilde{Y}_{k+1} = (y_k, y_{k-1}, \ldots, y_{k-n+1})$ and $\tilde{S}_{k+1} = (s_k, s_{k-1}, \ldots, s_{k-n+1})$. Quoting Ortega and Rheinboldt (1970) "...(sequential methods) are prone to unstable behavior and ... no satisfactory convergence results can be given". Nevertheless Gragg and Stewart (1976) propose a method which avoid instabilities by working with orthogonal factorizations of the involved matrices. Martinez (1979) gives three implementations of the idea proposed by Gragg and Stewart (1976) and some numerical experiments.

Multi-step quasi-Newton methods have been proposed by Moghrabi (1993), Ford and Moghrabi (1997) and Ford (1999) in the context of nonlinear programming. An interpolating path is built based on previous iterates, and used to produce an alternative secant equation. Interestingly, the best numerical results were obtained with no more than two steps.

We believe that the comments about the poor numerical stability of those methods found in major reference texts such as Dennis and Schnabel (1996) and Ortega and Rheinboldt (1970) have not encouraged more researchers to investigate generalizations of Broyden's method. We provide here such a generalization with robust properties and exhibiting an excellent behavior on numerical examples.

4.3 Population-based generalization

We propose here a class of methods calibrating a linear model based on several previous iterates. The difference with existing approaches is that we do not

impose the linear model to interpolate the function. Instead, we prefer to identify the linear model which is as close as possible to the nonlinear function, in the least-squares sense.

At each iteration, we maintain a finite population of previous iterates. Without loss of generality, we present the method assuming that all previous iterates $x_0, \ldots, x_{k+1}$ are considered. Our method belongs also to the quasi-Newton framework defined by (4.3), where B_{k+1} is computed using the following least squares:

$$B_{k+1} = \underset{J}{\operatorname{argmin}} \left(\sum_{i=0}^{k} \left\| \omega_{k+1}^{i} F(x_i) - \omega_{k+1}^{i} L_{k+1}(x_i; J) \right\|_F^2 \right) \tag{4.10}$$

where L_{k+1} is defined by 4.2. The weights $\omega_{k+1}^{i} \in \mathbb{R}^+$ capture the relative importance of each iterate. If $k < n$ the least-squares problem (4.10) is underdetermined. To overcome this obstacle we prefer to pose:

$$B_{k+1} = \underset{J}{\operatorname{argmin}} \left(\sum_{i=0}^{k} \left\| \omega_{k+1}^{i} F(x_i) - \omega_{k+1}^{i} L_{k+1}(x_i; J) \right\|_F^2 + \left\| \Gamma J - \Gamma B_{k+1}^{0} \right\|_F^2 \right) \tag{4.11}$$

where $\widehat{B}_{k+1}^{0} \in \mathbb{R}^{n \times n}$ is an a priori approximation of B_{k+1}. The role of the second term is to overcome the under-determination of the least-squares problem based on the first term and also control the numerical stability of the method. The matrix Γ contains weights associated with the arbitrary term $\widehat{B}_{k+1}^{0}$, and the weights $\omega_{k+1}^{i} \in \mathbb{R}^+$ are associated with the previous iterates. Equation (4.11) can be written in matrix form as follows:

$B_{k+1} =$

$$\underset{J}{\operatorname{argmin}} \left\| J \begin{pmatrix} S_{k+1} & I_{n\times n} \end{pmatrix} \begin{pmatrix} \Omega & 0_{k\times n} \\ 0_{n\times k} & \Gamma \end{pmatrix} - \begin{pmatrix} Y_{k+1} & \widehat{B}_{k+1}^{0} \end{pmatrix} \begin{pmatrix} \Omega & 0 \\ 0 & \Gamma \end{pmatrix} \right\|_F^2$$

where $\Omega \in \mathbb{R}^{k+1}$ is a diagonal matrix with weights ω_{k+1}^{i} on the diagonal for $i = 0, \cdots, k$. The normal equations of the least-squares problem lead to the following formula:

$$B_{k+1} = \widehat{B}_{k+1}^{0} + \left(Y_{k+1} - \widehat{B}_{k+1}^{0} S_{k+1} \right) \Omega^2 S_{k+1}^{T} \left(\Gamma^2 + S_{k+1} \Omega^2 S_{k+1}^{T} \right)^{-1}, \tag{4.12}$$

where $y_i = F(x_{k+1}) - F(x_i)$ and $s_i = x_{k+1} - x_i$ for $i = 0, \ldots, k$, $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_0)$ and $S_{k+1} = (s_k, s_{k-1}, \ldots, s_0)$.

The role of the a priori matrix $\widehat{B}_{k+1}^{0}$ is to overcome the possible under-determination of problem (4.11). For example, choosing $\widehat{B}_{k+1}^{0} = B_k$ (similarly to classical Broyden-like methods) exhibits good properties. In that case, (4.12) becomes an update formula, and local convergence can be proved (see Section 4.4).

The weights ω_{k+1}^{i} capture the relative importance of each iterate in the population. Roughly speaking, they should be designed in the lines of the

assumptions of Taylor's theorem, that is assigning more weight to points close to x_{k+1}, and less weight to points which are faraway. The matrix Γ captures the importance of the arbitrary terms defined by $\widehat{B}^0_{k+1}$ for the identification of the linear model. The weights have to be finite, and Γ must be such that

$$\Gamma^2 + S_{k+1}\Omega^2 S_{k+1}^T \tag{4.13}$$

is safely positive definite. Clearly $S_{k+1}\Omega^2 S_{k+1}^T$ is theoretically positive semidefinite. But zero or very small eigenvalues of matrices (4.13) are problematic for the numerical stability of the method. To ensure this property we describe below three possible approaches for choosing Γ^2: the *geometrical approach*, based on specific geometric properties of the population, the *subspace decomposition* approach, decomposing $\mathbb{R}^n$ into the subspace spanned by the columns of S_{k+1} and its orthogonal complement, and the *numerical approach*, designed to guarantee a numerically safe positive definiteness of (4.13).

4.3.1 Geometrical approach

The *geometrical approach* assumes that $n+1$ members of the population form a simplex, so that the columns of S_{k+1} span $\mathbb{R}^n$, and (4.13) is positive definite with $\Gamma = 0$. In that case, (4.12) becomes

$$B_{k+1} = Y_{k+1}\Omega^2 S_{k+1}^T \left(S_{k+1}\Omega^2 S_{k+1}^T\right)^{-1}. \tag{4.14}$$

If there are exactly $n+1$ iterates forming a simplex, the geometrical approach is equivalent to the interpolation method proposed by Wolfe (1959), and (4.14) is exactly (4.9), as S_{k+1} is square and non singular in that case. This approach have not shown good numerical behavior in practice as mentioned in Section 4.2. Also, it requires at least $n+1$ iterates, and may not be appropriate for large-scale problems.

4.3.2 Subspace decomposition

The *subspace decomposition* approach is based on the QR decomposition of S_{k+1}. We denote by r the rank of S_{k+1}, with $r \leq n$, and we have $S_{k+1} = QR$, where

$$Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \tag{4.15}$$

with Q_1 is $(n \times r)$, Q_2 is $(n \times n-r)$, and R is $(n \times k)$. The r columns of Q_1 form an orthogonal basis of the range of S_{k+1}. We define now Γ such that

$$\Gamma^2 = \begin{pmatrix} 0_{r\times n} \\ Q_2^T \end{pmatrix} \tag{4.16}$$

that is Q^T where Q_1^T has been replaced by a null matrix. With this construction $\Gamma^2 + S_{k+1}\Omega^2 S_{k+1}^T$ is invertible and the weights associated with the arbitrary

matrix $\widehat{B}^0_{k+1}$ are null in the subspace generated by the columns of S_{k+1}. In the case where S_{k+1} spans the entire space then $r = n$ and Γ^2 is a null matrix. Consequently, (4.12) is equivalent to (4.14).

Numerical problems may happen when the columns of S_{k+1} are close to linear dependence. These are the problems already mentioned in the introduction, and reported namely by Ortega and Rheinboldt (1970) and Dennis and Schnabel (1996). Clearly, such problems do not occur when S_{k+1} has exactly one column, which leads to the classical Broyden method.

Generalization of the Broyden good method

With the subspace decomposition approach, the changes of F predicted by B_{k+1} in a direction orthogonal to the range of S_{k+1} is the same as the one predicted by the arbitrary matrix $\widehat{B}^0_{k+1}$. This idea is exactly the same as the one used by Broyden (1965) to construct his so called *Broyden's good method*. Actually, the classical Broyden update (4.6) is a special case of our update formula (4.12), if $\widehat{B}^0_{k+1} = B_k$ and the population contains just two iterates x_k and x_{k+1}. The secant equation (4.5) completely defines the linear model in the one-dimensional subspace spanned by $s_k = x_{k+1} - x_k$, while an arbitrary decision is made for the rest of the model. If we define $\omega^k_{k+1} = 1$ and Γ^2 is given by (4.16), we can write (4.12) as

$$B_{k+1} = B_k + (y_k - B_k s_k)\, s_k^T \left(\Gamma^2 + s_k s_k^T\right)^{-1}. \tag{4.17}$$

The equivalence with (4.6) is due to the following equality

$$s_k^T \left(\Gamma^2 + s_k s_k^T\right)^{-1} = s_k^T \frac{1}{s_k^T s_k}, \tag{4.18}$$

obtained from the fact that $s_k^T \Gamma^2 = 0$, by (4.16).

4.3.3 Numerical approach

The *numerical approach* is designed to address both the problem of overcoming the under-determination, and of guaranteeing numerical stability. It is directly inspired by the modified Cholesky factorization proposed by Schnabel and Eskow (1991). The modified Cholesky factorization of a square matrix A creates a matrix E such that $A+E$ is safely positive definite, while computing its Cholesky factorization. It may namely happen that A has full rank, but with smallest eigenvalue very small with regard to machine precision. In that case, E is non zero despite the fact that A is non singular. We apply this technique with $A = S_{k+1}\Omega^2 S_{k+1}^T$ and $E = \Gamma^2$. So, if the matrix $S_{k+1}\Omega^2 S_{k+1}^T$ is safely positive definite, $\Gamma^2 = 0$ and (4.12) reduces to (4.14). If not, the modified Cholesky factorization guarantees that the role of the arbitrary term Γ is minimal.

We conclude this section by emphasizing important advantages of our generalization combined with the *numerical approach*. Firstly, contrarily to interpolation methods, our least-squares model allows to use more than p points to identify a model in a subspace of dimension p (where $p \leq n$). This is very important when the objective function is expensive to evaluate. Indeed, we make an efficient use of all the available information about the function to calibrate the secant model. It is namely advantageous compared to Broyden's method, where only two iterates are used to build the model. Secondly, the numerical approach proposed above controls the numerical stability of the model construction process, when a sequence of iterates may be linearly dependent. Finally, the fact that existing methods are special cases of our approach allows to exploit all the theoretical and practical properties already published in the literature, and simplifies their extension to our context. We apply this principle is the local convergence analysis in the next section. The main drawback is the increase in numerical linear algebra as the least-squares problem (4.11) must be solved at each iteration.

4.4 Local convergence analysis

We show that if Γ^2 is determined by the numerical approach described in Section 4.3, algorithm (4.3), where B_{k+1} is defined by (4.12) in his update form (i.e. $\widehat{B}^0_{k+1} = B_k$), locally converges to a solution of (4.1) if the following assumptions are verified.

Assumptions on the problem:

(P1) $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable in an open convex set $\mathcal{D}$.

(P2) The system of equations has a solution, that is $\exists x^* \in \mathcal{D}$ such that $F(x^*) = 0$.

(P3) $J(x)$ is Lipschitz continuous at x^* with constant K_{lip}, that is

$$\|J(x) - J(x^*)\| \leq K_{lip}\|x - x^*\| \quad \forall x \in \mathcal{D}. \tag{4.19}$$

in the neighborhood $\mathcal{D}$.

(P4) $J(x^*)$ is non-singular and we define $\gamma > 0$ such that $\|J(x^*)^{-1}\| < \gamma$.

Assumptions on the algorithm:

(A1) The algorithm is based on the iteration (4.3) with x_0 and B_0 as initial guess.

(A2) B_k is generated by (4.12) with $\widehat{B}^0_{k+1} = B_k$.

(A3) Γ^2 is computed using the *numerical approach*.

(A4) $\forall i \leq k$, we have $\omega_{k+1}^i \leq M_\omega$ for all k where $M_\omega \in \mathbb{R}^+$.

(A5) The size of the population $\mathcal{P}$ is bounded above by $M_\mathcal{P} \in \mathbb{N}$.

The notation $\|\cdot\|$ is used for the l_2 vector norm $\|x\| = (x^T x)^{\frac{1}{2}}$, and for any matrix norm which is consistent with the l_2 norm in the sense that $\|Ax\| \leq \|A\|\|x\|$ for each $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n\times n}$. In particular, the l_2 matrix norm and the Frobenius norm are consistent with the l_2 vector norm. For the sake of simplification, we denote $\omega_{k+1}^i = \omega_i$, $S = S_{k+1}$, $Y = Y_{k+1}$ and $I_p = \{1, \ldots, p\}$. The proof uses some lemma. Lemma 3 and 4 are classical results from the literature and lemma 5–7 are technical lemma related to our method.

Lemma 3 *Let $F : \mathbb{R}^n \longrightarrow \mathbb{R}^n$ be continuously differentiable in the open convex $D \subset \mathbb{R}^n$, $x \in D$, and let J be Lipschitz continuous at x in the neighborhood D with constant K_{lip}. Then for any $u, v \in D$,*

$$\|F(v) - F(u) - J(x)(v-u)\| \leq K_{lip}\frac{\|v-x\| + \|u-x\|}{2}\|v-u\|. \tag{4.20}$$

Proof. See, for example, Dennis and Schnabel, 1996. □

Lemma 4 *Let $A, C \in \mathbb{R}^{n\times n}$ and assume that A is invertible, with $\left\|A^{-1}\right\| \leq \mu$. If $\|A - C\| \leq \beta$ and $\beta\mu < 1$, then C is also invertible and*

$$\left\|C^{-1}\right\| \leq \frac{\mu}{1-\beta\mu}. \tag{4.21}$$

Proof. This lemma is known as the Banach Perturbation Lemma. (See, for example, Ortega and Rheinboldt, 1970). □

Lemma 5 *If assumptions (A4)-(A5) are verified, then*

$$\|S\Omega^2 S^T\| \leq 2M_\mathcal{P} M_\omega^2 \max_{i\in I_{k+1}} \|x_i - x^*\|^2, \tag{4.22}$$

$$\|\Omega^2 S^T\| \leq \sqrt{2M_\mathcal{P}} M_\omega^2 \max_{i\in I_{k+1}} \|x_i - x^*\|, \tag{4.23}$$

where x^* is solution of (4.1).

Proof.

$$\begin{aligned}
\|S\Omega^2 S^T\| &\leq \|S\Omega\|^2 && (4.24)\\
&\leq \sum_{i=1}^{k} \|\omega_i s_i\|^2 && (4.25)\\
&\leq k \max_{i\in I_k} (|\omega_i|\|s_i\|)^2 && (4.26)\\
&\leq k \max_{i\in I_k} (|\omega_i|\|x_{k+1} - \tilde{x} + \tilde{x} - x_i\|)^2 && (4.27)\\
&\leq 2k \max_{i\in I_k} |\omega_i|^2 \max_{i\in I_{k+1}} \|x_i - \tilde{x}\|^2 && (4.28)\\
&\leq 2M_\mathcal{P} M_\omega^2 \max_{i\in I_{k+1}} \|x_i - \tilde{x}\|^2 && (4.29)
\end{aligned}$$

for all $\tilde{x} \in \mathbb{R}^{n\times n}$, in particular with $\tilde{x} = x^*$ which proves (4.22).

$$\begin{aligned}
\|\Omega^2 S^T\|^2 &\leq \sum_{i=1}^{k} \|\omega_i^2 s_i\|^2 && (4.30)\\
&\leq k \max_{i\in I_k} \left(|\omega_i|^2 \|s_i\|\right)^2 && (4.31)\\
&\leq k \max_{i\in I_k} |\omega_i|^4 \max_{i\in I_k} \|x_{k+1} - \tilde{x} + \tilde{x} - x_i\|^2 && (4.32)\\
&\leq 2k \max_{i\in I_k} |\omega_i|^4 \max_{i\in I_{k+1}} \|x_i - \tilde{x}\|^2 && (4.33)
\end{aligned}$$

for all $\tilde{x} \in \mathbb{R}^{n\times n}$. We obtain (4.23) with $\tilde{x} = x^*$:

$$\|\Omega^2 S^T\| \leq \sqrt{2M_{\mathcal{P}}} M_\omega^2 \max_{i\in I_{k+1}} \|x_i - x^*\| \quad (4.34)$$

□

Lemma 6 *If assumptions (P1),(P2) and (P3) are verified then:*

$$\|(Y - J(x^*)S)\| \leq \sqrt{2M_{\mathcal{P}}} K_{lip} \max_{i\in I_{k+1}} \left(\|x_i - x^*\|^2\right) \quad (4.35)$$

where x^* is solution of (4.1).

Proof. Writing explicitly a column of the matrix $A = Y - J(x^*)S$

$$a_{\cdot j} = F(x_{k+1}) - F(x_i) - J(x^*)(x_{k+1} - x_i) \quad (4.36)$$

with $a_{\cdot j}$ defining the column j of $A = (a_{ij})$.

Using (4.36) and Lemma 3 we can write:

$$\begin{aligned}
\|Y - J(x^*)S\|^2 &\leq \sum_{j=1}^{k} \|a_{\cdot j}\|^2 && (4.37)\\
&\leq k \max_{i\in I_k} \|F(x_{k+1}) - F(x_i) - J(x^*)(x_{k+1} - x_i)\|^2 && (4.38)\\
&\leq k K_{lip}^2 \max_{i\in I_k} \left(\frac{\|x_i - x^*\| - \|x_{k+1} - x^*\|}{2} \|x_{k+1} - x_i\|\right)^2 && (4.39)\\
&\leq 2k K_{lip}^2 \max_{i\in I_{k+1}} \|x_i - x^*\|^2 \max_{i\in I_{k+1}} \|x_i - x^*\|^2 && (4.40)
\end{aligned}$$

Taking the square root on both side:

$$\|Y - J(x^*)S\| \leq \sqrt{2M_{\mathcal{P}}} K_{lip} \max_{i\in I_{k+1}} \|x_i - x^*\|^2 \quad (4.41)$$

□

Lemma 7 *If assumption (A3) is verified, then*

$$\| \left(\Gamma^2 + S\Omega^2 S^T \right)^{-1} \| \leq \frac{1}{\tau} \tag{4.42}$$

where $\tau > 0$.

Proof. Let $A \in \mathbb{R}^{n \times n}$, we denote by $\lambda_m(A)$ and $\lambda_M(A)$ its smallest and largest eigenvalues, respectively. So we can write using the definition of the l_2 norm:

$$\begin{aligned} \|(\Gamma^2 + S\Omega^2 S^T)^{-1}\| &= \lambda_M((\Gamma^2 + S\Omega^2 S^T)^{-1}) & (4.43) \\ &= \frac{1}{\lambda_m(\Gamma^2 + S\Omega^2 S^T)}. & (4.44) \end{aligned}$$

From assumption (A3), Γ^2 is computed using the modified Cholesky factorization, proposed by Schnabel and Eskow (1991), with parameter τ. Therefore,

$$\lambda_m(\Gamma^2 + S\Omega^2 S^T) \geq \tau, \tag{4.45}$$

which concludes the proof. □

The parameter τ in Lemma 7 controls the way we perturb $S\Omega^2 S^T$. It guarantees that the smallest eigenvalue of $\Gamma^2 + S\Omega^2 S^T$ is strictly greater than τ and, therefore, safely positive in a finite arithmetic context if τ is properly chosen. Schnabel and Eskow (1991) suggest to choose $\tau = (\text{macheps})^{\frac{1}{3}}$ where macheps is the machine epsilon.

Theorem 8 *Let assumptions (P1) to (P3) hold for the problem and assumptions (A1) to (A5) hold for the algorithm. Then there exist two non-negative constants* α_1 *and* α_2 *such that for each* x_k *and* B_k*:*

$$\begin{aligned} \|B_{k+1} - J(x^*)\| \leq & \left(1 + \alpha_1 \max_{i \in I_{k+1}} \|x_i - x^*\|^2\right) \|B_k - J(x^*)\| \\ & + \alpha_2 \max_{i \in I_{k+1}} \|x_i - x^*\|^3 . \end{aligned} \tag{4.46}$$

Proof. From the update formula (4.12), and defining

$$\begin{aligned} T_1 &= I - S\Omega^2 S^T (\Gamma^2 + S\Omega^2 S^T)^{-1} \\ T_2 &= (Y - J(x^*)S)\Omega^2 S^T (\Gamma^2 + S\Omega^2 S^T)^{-1}, \end{aligned}$$

we obtain

$$\begin{aligned} \|B_{k+1} - J(x^*)\| &= \|B_k - J(x^*) + [(J(x^*)S - J(x^*)S) + (Y - B_k S)]\, \Omega^2 S^T (\Gamma^2 + S\Omega^2 S^T)^{-1}\| \\ &\leq \|T_1\| \|B_k - J(x^*)\| + \|T_2\|. \end{aligned}$$

From Lemma 5 and 7 we obtain

$$\begin{aligned} \|T_1\| &\leq \|I\| + \|S\Omega^2 S^T\| \|(\Gamma^2 + S\Omega^2 S^T)^{-1}\| & (4.47) \\ &\leq 1 + \alpha_1 \max_{i \in I_{k+1}} \|x_i - x^*\|^2, & (4.48) \end{aligned}$$

with $\alpha_1 = \frac{2}{\tau} M_{\mathcal{P}} M_{\omega}^2 > 0$. We conclude the proof using Lemma 5, 6 and 7 to show that:

$$\begin{aligned} \|T_2\| &\leq \|(Y - J(x^*)S)\| \|\Omega^2 S^T\| \|(\Gamma^2 + S\Omega^2 S^T)^{-1}\| && (4.49) \\ &\leq \alpha_2 \max_{i \in I_{k+1}} \|x_i - x^*\|^3, && (4.50) \end{aligned}$$

with $\alpha_2 = \frac{2}{\tau} K_{lip} M_{\mathcal{P}} M_{\omega}^2 > 0$. □

Theorem 9 *Let assumptions (P1) to (P3) hold for the problem and assumptions (A1) to (A5) hold for the algorithm. If, for any* $r \in]0,1[$ *there exists* $\varepsilon(r)$ *and* $\delta(r)$ *such that*

$$\|x_0 - x^*\| \leq \varepsilon(r) \tag{4.51}$$

and

$$\|B_0 - J(x^*)\| \leq \delta(r) \tag{4.52}$$

then the sequence $x_{k+1} = x_k - B_k^{-1} F(x_k)$ *is well defined and converges* q*-linearly to* x^* *with* q*-factor at most* r. *Furthermore, the sequences* $\{\|B_k\|\}_k$ *and* $\left\{\|B_k^{-1}\|\right\}_k$ *are uniformly bounded.*

Proof. The structure of the demonstration is similar to the proof of Theorem 3.2 in Broyden et al. (1973). We have purposely skipped some identical technical details.

First choose $\varepsilon(r) = \varepsilon$ and $\delta(r) = \delta$ such that

$$\gamma(1 + r)\,(K_{lip}\varepsilon + 2\delta) \leq r \tag{4.53}$$

and

$$\left(2\alpha_1 + \alpha_2 \frac{\epsilon}{1 - r}\right) \frac{\epsilon^2}{1 - r^2} \leq \delta. \tag{4.54}$$

We invoke Lemma 4 with $\mu = \gamma$ and $\beta = 2\delta$ to prove that B_0 is non-singular and

$$\|B_0^{-1}\| < \gamma(1 + r). \tag{4.55}$$

Note that assumption $2\delta\gamma < 1$ for Lemma 4 is directly deduced from (4.53).

The improvement after the first iteration, that is

$$\|x_1 - x^*\| \leq r\|x_0 - x^*\|, \tag{4.56}$$

is deduced from lemma 3. Indeed,

$$\begin{aligned} \|x_1 - x^*\| &\leq \|x_0 - B_0^{-1} F(x_0) - x^*\| \\ &\leq \|B_0^{-1}\,(B_0 x_0 - F(x_0) + F(x^*) - B_0 x^* - J^*(x_0 - x^*) + J^*(x_0 - x^*))\,\| \\ &\leq \|B_0^{-1}\|\,(\|(B_0 - J^*)(x_0 - x^*) - F(x_0) - F(x^*) - J^*(x_0 - x^*)\|) \\ &\leq \|B_0^{-1}\|\,(\|F(x_0) - F(x^*) - J^*(x_0 - x^*)\| + \|B_0 - J^*\| \|x_0 - x^*\|) \\ &\leq (1 + r)\gamma\,(K_{lip} \max(0, \|x_0 - x^*\|)\|x_0 - x^*\| + 2\delta\|x_0 - x^*\|) \\ &\leq (1 + r)\gamma\,(K_{lip}\varepsilon + 2\delta)\,\|x_0 - x^*\|, \end{aligned}$$

and (4.56) is verified using (4.53).

The result for iteration k is proven with an induction argument based on the following recurrence assumptions:

$$\|B_m - J^*\| \leq 2\delta \tag{4.57}$$
$$\|x_{m+1} - x^*\| \leq r\|x_m - x^*\| \tag{4.58}$$

for all $m = 1, \ldots, k-1$.

We first prove that $\|B_k - J^*\| \leq 2\delta$ using Theorem 8. From (4.46) we deduce $\|B_{m+1} - J(x^*)\| - \|B_m - J(x^*)\|$

$$\begin{aligned} &\leq \alpha_1 \max_{i \in I_{m+1}} \|x_i - x^*\|^2 \|B_m - J(x^*)\| + \alpha_2 \max_{i \in I_{m+1}} \|x_i - x^*\|^3 \\ &\leq \alpha_1 r^{2(m+1)} \varepsilon^2 2\delta + \alpha_2 r^{3(m+1)} \varepsilon^3 \end{aligned} \tag{4.59}$$

Summing both sides of (4.59) for m ranging from 0 to $k-1$, we deduce that

$$\|B_k - J(x^*)\| \leq \|B_0 - J(x^*)\| + \left(2\alpha_1\delta + \alpha_2 \frac{\varepsilon}{1-r}\right) \frac{\varepsilon^2}{1-r^2} \tag{4.60}$$
$$\leq 2\delta, \tag{4.61}$$

where (4.61) derives from (4.52) and (4.54).

The fact that B_k is invertible and $\|B_k^{-1}\| \leq \gamma(1+r)$ is again a direct application of the Banach Perturbation Lemma 4.

To prove (4.58) for $m = k$ we use the same development as (4.56). In brief, using lemma 3 we obtain:

$$\begin{aligned} \|x_{m+1} - x^*\| &\leq \|x_m - B_m^{-1} F(x_m) - x^*\| \\ &\leq \|B_m^{-1}\| \left(\|F(x_m) - F(x^*) - J^*(x_m - x^*)\| + \|B_m - J^*\| \|x_m - x^*\|\right) \\ &\leq (1+r)\gamma \left(K_{lip}\varepsilon + 2\delta\right) \|x_m - x^*\| \\ &\leq r\|x_m - x^*\|, \end{aligned}$$

concluding the induction proof. □

4.5 Performance evaluation

We present here a preliminary analysis of the performance of our method, in comparison with classical algorithms. All algorithms and test functions have been implemented with the package Octave (see `http://www.octave.org/`) and computations have been done on a laptop equipped with 1066MHz CPU in double precision. The machine epsilon is about 2.2204e-16.

4.5.1 Problems

The numerical experiments were carried out on a set of 30 test functions, seven with fix dimension and 23 with dimension $n = 6, 10, 20, 50, 100$. This set is composed of the four standard nonlinear systems of equations proposed by Dennis and Schnabel (1996): (that is, *Extended Rosenbrock Function*, *Extended Powell Singular Function*, *Trigonometric Function*, *Helical Valley Function*), three functions from Broyden (1965), five functions proposed by Kelley (2002b) in his new book on Newton's method (that is, *Arctangent Function*, *a Simple Two-dimensional Function*, *Chandrasekhar H-equation*, *Ornstein - Zernike Equations*, *Right Preconditioned Convection-Diffusion Equation*), the test functions given by Spedicato and Huang (1997) and three linear systems of equations of the form $Ax = b$. They have been designed to challenge the tested algorithms.

1. For the first, the matrix A is the Hilbert matrix, and vector b is composed of all ones.

2. The second problem is based on the matrix A such that $a_{ij} = j$ if $i + j = n + 1$, and $a_{ij} = 0$ otherwise. All entries of the right-hand side b are -10. Its structure is designed so that the identity matrix is a poor approximation.

3. The third problem is based on a Vandermond matrix $A(v)$ with $v = (-1, -2, \ldots, -n)$. All entries of the right-hand side b are -1.

The starting point for all those problems is $x = (1, \ldots, 1)^T$.

For each problem, we have used the starting point proposed in the original paper. In order to challenge even more the algorithms, we have also considered $10x_0$ as another starting point for each problem. The total experimental design is therefore composed of 244 problems. The results include only problems which have been solved at least by one of the considered methods, that is 124 problems.

4.5.2 Methods

The algorithms that have been considered are based upon the undamped quasi-Newton method, i.e. without any step control or globalization methods. This allows us to compare their speed of convergence, in term of number of function evaluations, and their robustness without introducing a bias due to the step control or the globalization method. All the algorithms have the following structure:

- Given $F : \mathbb{R}^n \to \mathbb{R}^n$, $x_0 \in \mathbb{R}^n$ and $B_0 \in \mathbb{R}^{n \times n}$
- While stopping criteria is not verified:
  - Find s solving $B_k s = -F(x_k)$,

- Evaluate $F(x_{k+1})$ where $x_{k+1} = x_k + s$,
- Compute B_{k+1},

with the following characteristics: the initial Jacobian approximation B_0 is the same for all algorithms and equal to the identity matrix. The stopping criteria is a composition of three conditions: small residual, that is $\|F(x_k)\|/\|F(x_0)\| \leq 10e{-}6$, maximum number of iterations ($k \geq 200$ for problems of size $n \leq 20$ and $k \geq 500$ for problems of size $n = 50$ and $n = 100$), and divergence, diagnosed if $\|F(x_k)\| \geq 10e10$.

Consequently, the algorithms differ only by the method to compute B_{k+1}. We consider four of them:

1. Broyden's good Method (BgM), using the update (4.6).

2. Broyden's bad Method (BbM), also proposed by Broyden (1965). It is based on the following secant equation:

$$s_k = B_{k+1}^{-1} y_k. \tag{4.62}$$

   and directly computes the inverse of B_k:

$$B_{k+1}^{-1} = B_k^{-1} + \frac{(s_k - B_k^{-1} y_k)\, y_k^T}{y_k^T y_k}. \tag{4.63}$$

   Broyden (1965) describes this method as "bad", that is numerically unstable. However, we have decided to include it in our tests for the sake of completeness. Moreover, as discussed in Chapter 5, it does not always deserve its name.

3. The Hybrid Method (HMM) proposed by Martinez (1982). At each iteration, the algorithm decides to apply either BgM or BbM. Martinez (2000) observes a systematic improvement of the Hybrid approach with respect to each individual approach. As discussed below, we obtain the same result.

4. Our Generalized Secant Method (GSM), defined by (4.12) in its update form with $\widehat{B}_{k+1}^0 = B_k$ using the *numerical approach* described in Section 4.3, with $\tau = (\text{macheps})^{\frac{1}{3}}$. Contrarily to what is presented in the theoretical analysis, we consider only $p = \max(n, 10)$ previous iterates in the population. Indeed, including all previous iterates may generate memory management problems, and anyway does not significantly affect the behavior of the algorithm. The weights are defined as

$$\omega_{k+1}^i = \frac{1}{\|x_{k+1} - x_i\|^2} \qquad \forall i \in I_p \tag{4.64}$$

Clearly, the comparative performance is problem dependent. Therefore, we use the performance profiles analysis methodology proposed by Dolan and More (2002), using the number of function evaluation as performance metric.

4.5.3 Performance profiles

The performance profile for a method is the cumulative distribution function for a given performance metric. If $f_{p,a}$ is the performance metric of algorithm a on problem p (the number of function evaluations in our case), then the *performance ratio* is defined by

$$r_{p,a} = \frac{f_{p,a}}{\min_a\{f_{p,a}\}}, \tag{4.65}$$

if algorithm a has converged for problem p, and $r_{p,a} = r_{\text{fail}}$ otherwise, where r_{fail} must be strictly larger than any performance ratio (4.65). For any given threshold π, the overall performance of algorithm a is given by

$$\rho_a(\pi) = \frac{1}{n_p}\Phi_a(\pi) \tag{4.66}$$

where n_p is the number of problems considered, and $\Phi_a(\pi)$ is the number of problems for which $r_{p,a} \leq \pi$.

In particular, the value $\rho_a(1)$ gives the probability that algorithm a wins over all other algorithms. The value $\lim_{\pi \to r_{\text{fail}}} \rho_a(\pi)$ gives the probability that algorithm a solves a problem and, consequently, provides a measure of the robustness of each method.

4.5.4 General results

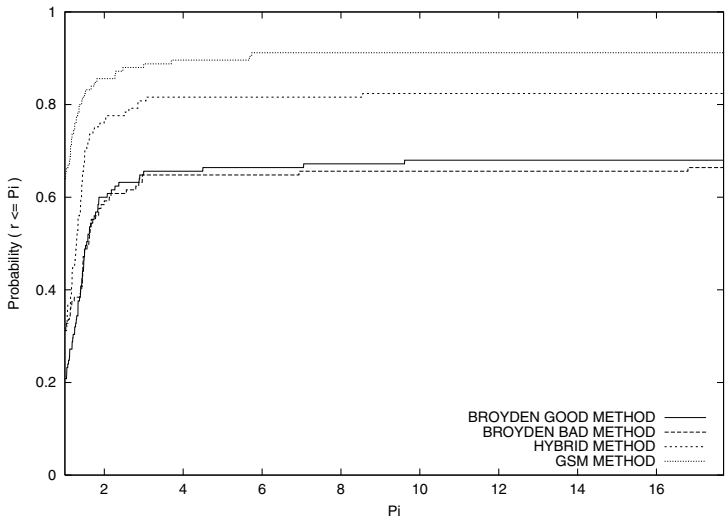


Figure 4.1: Performance profiles

We first present the performance profile for all algorithms described above on all problems on Figure 4.1. A zoom for π between 1 and 4 is provided in Figure 4.2.

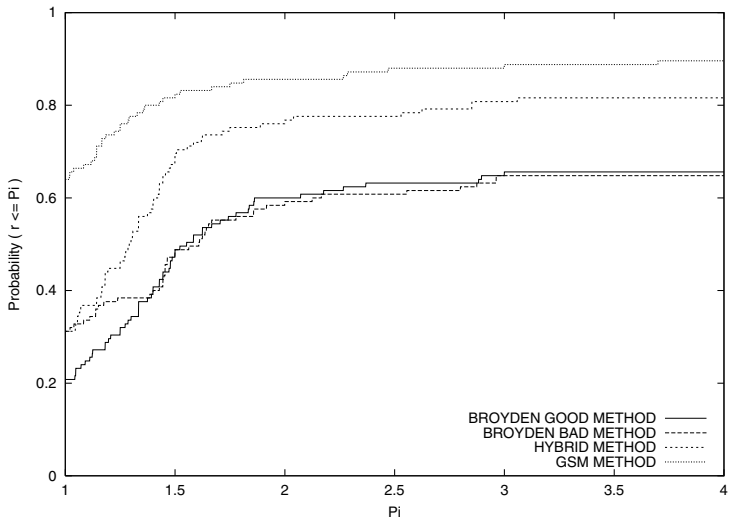


Figure 4.2: Performance profiles on (1,4)

The results are very satisfactory for our method. Indeed, we observe that GSM is best on nearly 70% of the problems, and is able to solve more than 90% of the problems. From Figure 4.2, we note also that when GSM is not the best method, it converges within a factor of 1.5 of the best algorithm for more than 80% of the problems. We also confirm results by Martinez (2000) showing that the Hybrid method is more reliable than BgM and BbM. Indeed, it converges on more than 80% of the problems, while each Broyden method converges only on 60% of the cases. Moreover, HMM wins as often as BbM does, but is more robust, as its performance profile grows faster than the profile for BbM. The relative robustness of BgM and BbM is comparable.

The performance profile analysis strongly depends on the number of methods that are being compared. Therefore, we like to present a comparison between BgM and GSM only, as BgM is probably the most widely used method. The significant improvement provided by our method over Broyden's method is illustrated by Figure 4.3, based on problems solved by at least one of the two methods, that is 112 problems.

4.5.5 Behavior in presence of noise

We conclude this section by a preliminary analysis of the behavior of our method in the presence of noise in the function. Indeed, we speculate that the use of a larger sample of iterates to calibrate the secant model smoothes the impact of noise on the method.

We consider a generalized noisy system of nonlinear equations described by:

$$G(x) = F_s(x) + \phi(x) \tag{4.67}$$

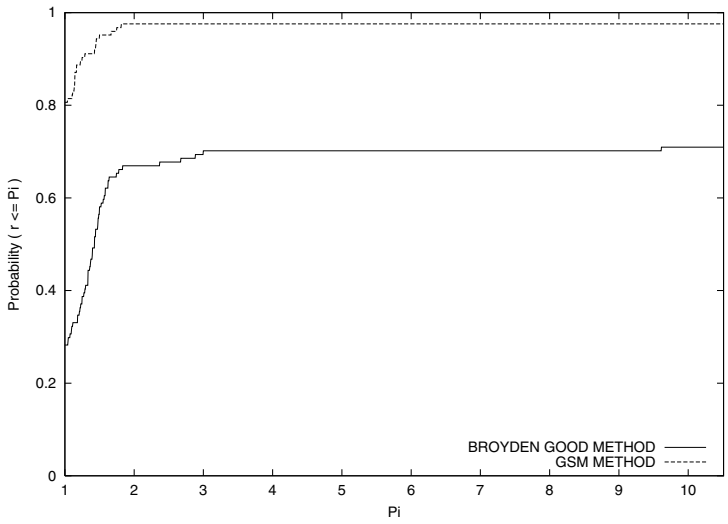


Figure 4.3: Performance profiles – BgM and GSM –

where F_s is a smooth system of nonlinear equations and $\phi(x)$ is a random perturbation. Similarly to Choi and Kelley (2000), we assume that the noise decreases near the solution, and we propose

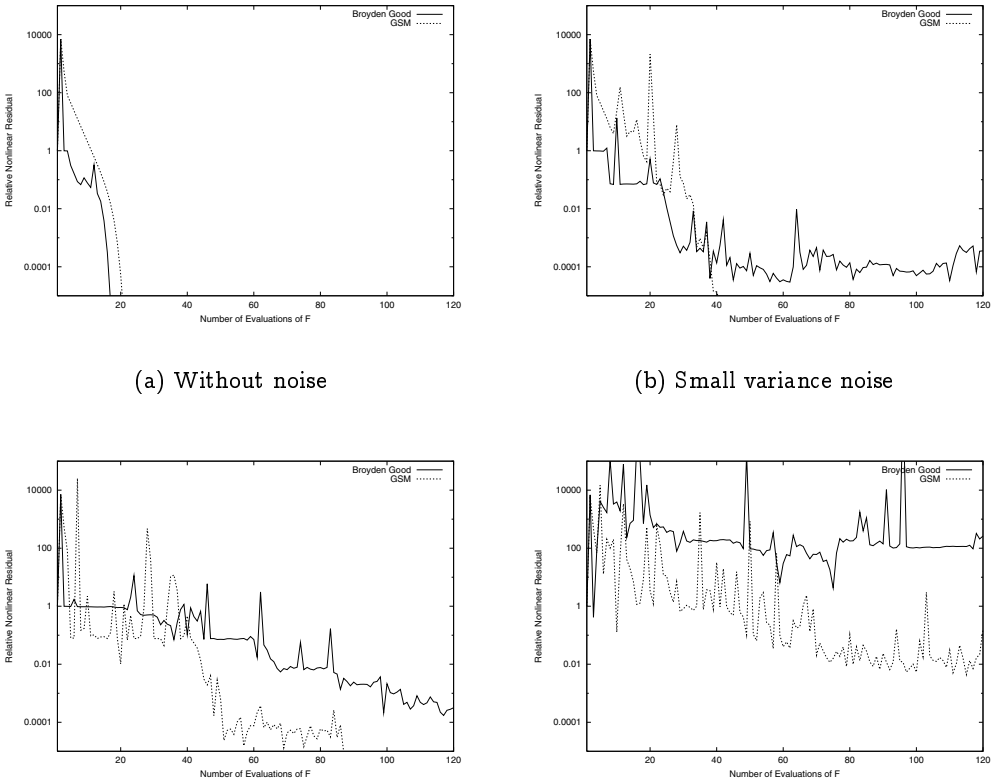
$$\phi(x) \sim N(0, \alpha^2 \|x - x^*\|^2). \tag{4.68}$$

We have selected a problem where BgM is better than GSM in the deterministic case. The considered problem is the following system of equations:

$$f_i = x_i - \frac{\sum_{j=1}^{4} x_j^3 + 1}{8} \qquad i = 1, \ldots, 4 \tag{4.69}$$

with initial point $x_0 = (1.5, \ldots, 1.5)$. The solution of this system is $x^* = (0.20432, \ldots, 0.20432)$. The results for 4 levels of stochasticity are presented in Figure 4.4. For each value of the parameter α in (4.68), we plot the relative nonlinear residual, that is $\|G(x_k)\|/\|G(x_0)\|$, against the number of function evaluations.

Figure 4.4(a) illustrates the deterministic case, with $\phi(x) = 0$, where BgM is slightly better than GSM. When a noise with small variance ($\alpha = 0.001$, Figure 4.4(b)) is present, GSM decreases the value of the residual pretty quickly, while the descent rate of BgM is much slower. When the variance of the noise increases ($\alpha = 0.05$ in Figure 4.4(c), and $\alpha = 1$ in Figure 4.4(d)), the BgM is trapped in higher values of the residual, while GSM achieves a significant decrease. We have performed the same analysis on other problems, and observed a similar behavior, that is a better robustness of GSM when solving a noisy system of equations.



(a) Without noise

(b) Small variance noise

(c) Normal variance noise

(d) Big variance noise

Figure 4.4: Behavior with stochasticity

4.6 Conclusion

We have proposed a new family of secant methods, based on the use of more than two iterates to identify the secant model. Contrarily to previous attempts for multi-iterate secant methods, the key ideas of this chapter are:

(i) to use a least-squares approach instead of an interpolation method to derive the secant model

(ii) to explicitly control the numerical stability of the method.

A specific instance of this family of methods provides an update formula. We have provided a proof of the local convergence of a quasi-Newton method based on this update formula. Moreover, we have performed numerical experiences with several algorithms. The results show that our method produces significant improvement in terms of robustness and number of function evaluations compared to classical methods quasi-Newton methods. Finally, we have provided preliminary evidences that our method is likely to be more robust in the presence of noise in the function.

The main drawback of our approach is the relatively high cost in numerical linear algebra, as a least-squares problem has to be solved at each iteration. This has not been addressed in this chapter, as only the number of function evaluations is under consideration, but it may become cumbersome for large problems. We deal with this problem is in the next chapter, where we analyze the following variants of these methods. First following Broyden's idea to derive BbM from (4.62), an update formula for B_{k+1}^{-1} can easily be derived in the context of our method. Secondly, non-update instances of our class of methods can be considered. In that case, the arbitrary matrix $\widehat{B}_{k+1}^{0}$ in (4.11) may be different from B_k. In that case, choosing a matrix independent from k would allow to apply Kalman filtering (Kalman, 1960) to incrementally solve our least squares (4.11) and, consequently, improve the numerical efficiency of the method. For large-scale problems, an iterative scheme such as LSQR (Paige and Saunders, 1982) can be considered, as proposed in Chapter 3. These ideas are the basis of the next chapter.

Finally, the ideas proposed in this chapter can easily be tailored to optimization problems, where the symmetry and positive definiteness of the matrix can be explicitly exploited.

Chapter 5

Large-scale systems

Contents

5.1 Introduction

Solving large-scale systems of nonlinear equations is becoming more and more a critical step in the analysis and management of complex systems. Namely, the increasing power of computers has motivated the development of simulation tools in many research areas. In general, these tools involve a large number of parameters, and are demanding in CPU time and memory. Often, only a limited number of evaluations is available to find a solution of the problem. We will refer to this limited number of function evaluations as a *computation budget*. For example, the Consistent Anticipatory Route Guidance problem described in Chapter 6 is designed to be used in a real-time dynamic traffic management system and, therefore, has a limited computation budget.

The most efficient approaches to solve systems of nonlinear equations are based on Newton's idea to model the nonlinear function with a tangent hyperplane, computed with the first-order derivatives. Clearly, if the function is expensive to compute, and the number of unknowns is large, the first-order derivatives are usually even more expensive, and sometimes not available, and Newton's idea cannot be directly used.

Two families of methods have been proposed to overcome the unavailability of the derivatives: *quasi-Newton* methods and *inexact Newton* methods. For large-scale problems, it seems that the most efficient methods belong to the inexact Newton family (Kelley, 2002a). This chapter describes an adaptation of GSM that inherits features from both families. These adaptations allow us to design a matrix-free algorithm, a crucial property for large-scale systems, and the concessions made to accommodate the large-scale characteristic do not deteriorate the robustness and the efficiency of GSM presented in chapter 4, leading to a very competitive algorithm to solve large-scale systems of nonlinear equations, even compared with methods using information about derivatives computed using finite differences.

This chapter is organized as follows. We start by describing existing methods to solve large-scale systems of nonlinear equations: large-scale quasi-Newton methods and inexact Newton methods. In section 5.3 we present the fundamental ideas of our new class of methods. In section 5.4 we formally describe a specific instance of our methods. Numerical results presented in Section 5.5 are divided into four parts. First we validate the intended adaptations showing that the performances of GSM, using the same set of test problems used in Chapter 4, are preserved. In section 5.5.2 we emphasize the efficiency and robustness of the proposed algorithm on classical large-scale systems of nonlinear equations compared with other methods designed to solve large-scale systems. We illustrate the impact of the parameters of our method on particular examples. Finally in Section 5.6 we show the reliability of our population-based approach on noisy problems compared to standard methods. Some conclusions and perspectives are outlined in Section 5.7.

5.2 Existing large-scale methods

Let F be a continuous mapping from $\mathbb{R}^n$ to $\mathbb{R}^n$ and consider the system of nonlinear equations:

$$F(x) = 0. \tag{5.1}$$

We assume that F has continuous partial derivatives on $\mathbb{R}^n$ and denote $\nabla F(x)^T = J(x)$ the Jacobian matrix. We are mostly interested in problems where n is large and the evaluations of F are expensive to compute, but without imposing any special structure of the Jacobian matrix.

5.2.1 Large-scale quasi-Newton methods

Quasi-Newton methods proceed as describes in Chapter 4. Given x_k, consider at each iteration the linear model (4.2) approximating $F(x)$ in the neighborhood of x_k and compute x_{k+1} as a solution of the linear system $L_k(x; B_k) = 0$. An iteration of a quasi-Newton method is described by:

$$\text{Solve } B_k p_k = -F(x_k) \tag{5.2}$$

$$\text{Define } x_{k+1} = x_k + p_k \tag{5.3}$$

followed by the computation of B_{k+1}. The way to update B_k differentiates essentially the various quasi-Newton methods, as Equations (5.2) and (5.3) determine entirely the algorithm. One way to compute the matrix B_k, as detailed in section 4.2, is to carry an approximation of the Jacobian satisfying at each iteration the equation (4.5). These methods are called secant methods.

Equation (4.5) ensures that the linear model interpolates $F(x)$ at x_k and x_{k+1}. The most successful class of secant methods has been proposed by Broyden (1965) and is defined by the update (4.6). This method has been nicknamed the Broyden "good" Method (BgM), in contrast with the Broyden "bad" Method (BbM) (Spedicato and Huang, 1997 and Martinez, 2000). This method, also proposed by Broyden (1965), is based on another secant equation:

$$H_{k+1} y_k = s_k \tag{5.4}$$

with $y_k = F(x_{k+1}) - F(x_k)$ and $s_k = x_{k+1} - x_k$ and where H_k is an approximation of $J(x_k)^{-1}$. In his seminal paper, Broyden (1965) writes that the method "appears in practice to be unsatisfactory", but without the denomination "bad". The origins of these nicknames are not clear, as described by Martinez in a private communication: "Historically, the two methods proposed by Broyden (1965) have been known as "first" and "second". Numerical experiments in the first years after their introduction suggested that the "good" method was far better than the "bad" method. After that, many authors began to call them "good" and "bad". In fact, I do not know exactly who was the first to publish those names, although I am sure that they were called in that way informally in meetings and congresses." Actually it turns out that the reasons why BgM

is "good" and BbM is "bad" are not well understood. Moreover there is no empirical evidence that BgM performs better than BbM, as pointed out by Martinez (2000) and illustrated by Figures 4.1 and 4.2. The BbM updates at each iteration an approximation of the inverse of the Jacobian, so that (5.2) becomes

$$p_k = -H_k F(x_k). \tag{5.5}$$

This avoids solving a linear system of equations, which is appropriate for large-scale problems. Broyden's update formula for BgM is

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)\, y_k^T}{y_k^T y_k}. \tag{5.6}$$

Other secant methods have been introduced with the aim of reducing the computational cost of Broyden's methods. The Column-Updating method (COLUM) has been introduced by Martinez (1984). At each iteration only one column of B_k is changed, in such a way that secant equation (4.5) is satisfied. This method has essentially the same behavior as the Broyden good method in terms of reliability and number of iterations (Friedlander et al., 1997).

Following Broyden, an Inverse Column-Updating Method (ICUM) has been introduced by Martinez and Zambaldi (1992). The update formula of ICUM is given by:

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k) e_{j_k}^T}{e_{j_k}^T y_k} \tag{5.7}$$

where j_k is such that $|e_{j_k}^T y_k| = \|y_k\|_\infty$. We easily remark that H_{k+1} is identical to H_k except on the j_k-th column. According to Spedicato and Huang (1997), Ruggiero et al. (1996), Luksan and Vlcek (1998) and Martinez (2000), this method is currently considered as one of the most effective secant methods to solve large-scale systems of nonlinear equations.

Broyden methods need to store B_k (or H_k). Therefore for high-dimensional systems, they might lead to memory constraints. Different techniques can be considered to modify and extend these methods in several ways to handle large-scale problems as described by Nocedal and Wright (1999). First limited-memory versions can be considered, where the matrices B_k (or H_k) are never explicitly generated (see Byrd et al., 1994). For instance, Gomes-Ruggiero et al. (1991) propose a limited-memory version of BgM. This allows to handle problems of larger dimensions, at the cost of a slight deterioration of the performance. Nocedal and Wright (1999) write "these methods are fairly robust, inexpensive and easy to implement, but they do not converge rapidly". Other approaches define quasi-Newton methods that preserve the sparsity of the true Jacobian or that can exploit the structural properties of the problem.

Most quasi-Newton methods for solving systems of nonlinear equations are local methods, meaning that they converge if the starting point is close enough to a solution. In many practical cases the domain of convergence of local methods is large enough to authorize good practical convergence properties. The

globalization of quasi-Newton methods, aiming at an algorithm converging from any starting point, is a complex issue (Martinez, 2000). Moreover, the robustness obtained through globalization is often obtained at the cost of efficiency (Ruggiero et al., 1996). Our own experience confirms these observations. We refer the reader to Dennis and Schnabel (1996) for an extensive analysis of Newton and quasi-Newton methods.

5.2.2 Inexact-Newton methods

Inexact Newton methods find an approximate solution of (5.2) with the help of an iterative algorithm. They identify a direction p_k satisfying the inexact Newton condition:

$$\|F(x_k) + J(x_k)p_k\| \leq \eta_k \|F(x_k)\| \tag{5.8}$$

for some $\eta_k \in [0, 1]$. The most conventional inexact Newton method uses iterative techniques to compute the Newton step p_k using (5.8) as a stopping criterion. Among these iterative techniques, Krylov-based linear solvers are generally chosen. This approach allows to perform only matrix-vector products, which is essential for large-scale problems as underlined by Kelley (2002b). Newton-Krylov methods need to estimate Jacobian-vector products using finite differences approximations in the appropriate Krylov subspace.

The iterative linear solver GMRES (Generalized Minimum RESidual) method proposed by Saad and Schultz (1986) computes at each iteration the solution x_k of the linear least-squares problem $\min_x \|b - Ax\|^2$ over the manifold S_k given by:

$$S_k = x_0 + \left\langle r_0, Ar_0, \ldots, A^{k+1}r_0 \right\rangle \tag{5.9}$$

We refer to Kelley (1995) for detailed descriptions of the algorithms. Despite their robustness and efficiency, Newton-Krylov methods are not appropriate for problems with noisy functions, where derivative-based methods must be avoided.

5.3 Fundamentals of the algorithm

We present here the general ideas of our method, starting from a quasi-Newton scheme. But contrarily to quasi-Newton methods, we do not construct the linear model. Similarly to inexact Newton methods, we directly compute a step p_k at each iteration.

More precisely, we propose an inverse version of the class of Generalized Secant Methods proposed in Chapter 4, updating directly the inverse of B_{k+1}. GSM uses the classical quasi-Newton framework (5.2)–(5.3) and calibrates, at each iteration, the matrix B_{k+1} with all previous iterates $x_0, \ldots, x_{k+1}$ solving the least-squares problem (4.11). Analogously, in the inverse version, we com-

pute x_{k+1} using (5.5) and (5.3), that is

$$\begin{aligned} p_k &= -H_k F(x_k) \\ x_{k+1} &= x_k + p_k. \end{aligned}$$

Following the ideas given in Section 4.3, we define H_{k+1} using a population of previous iterates $x_m, x_{m+1} \ldots, x_{k+1}$ where $m = \max(0, k - \kappa + 2)$, and κ is a user-defined parameter giving the maximum size of the population. Then,

$$H_{k+1} = \operatorname{argmin}_{J \in \mathbb{R}^{n \times n}} \sum_{i=m}^{k} \|\omega_{k+1}^i (J y_i - s_i)\|_F^2 = \|(J Y_{k+1} - S_{k+1})\,\Omega\|_F^2 \quad (5.10)$$

where $y_i = F(x_{k+1}) - F(x_i)$ and $s_i = x_{k+1} - x_i$ for $i = m, \ldots, k$, $Y_{k+1} = (y_k, y_{k-1}, \ldots, y_m)$, $S_{k+1} = (s_k, s_{k-1}, \ldots, s_m)$ and Ω is a diagonal matrix with weights ω_{k+1}^i on the diagonal.

The weights $\omega_{k+1}^i \in \mathbb{R}$ capture the relative importance of each iterate of the population. Roughly speaking, they should be designed in the lines of the assumptions of Taylor's theorem, that is assigning more weight to points close to $F(x_{k+1})$, and less weight to points that are far away. The choice of these weights as well as the selection of the size of the population κ are discussed in Section 5.4.

Clearly, if the size of the population is less than $n + 1$, the least-squares problem (5.10) is underdetermined. This immediately occurs if $\kappa < n + 1$, a common case for large-scale problems. We introduce an a priori target matrix $\widehat{H}_{k+1}^0$, and extend the least-squares problem as follows.

$$H_{k+1} = \operatorname*{argmin}_J \left\| J(Y_{k+1}\Omega \quad \Gamma) - (S_{k+1}\Omega \quad \Gamma \widehat{H}_{k+1}^0) \right\|_F^2. \quad (5.11)$$

where $\Gamma \in \mathbb{R}^{n \times n}$ is diagonal and captures the relative importance of the arbitrary term $\widehat{H}_{k+1}^0$. The choice of Γ is discussed in Section 5.4.

Obviously, (5.11) is intractable for large-scale problems. We present now a procedure that solves this problem implicitly. To simplify the notations, let

$$A = (Y_{k+1}\Omega \quad \Gamma) \quad (5.12)$$

$$C = \left(S_{k+1}\Omega \quad \Gamma \widehat{H}_{k+1}^0\right), \quad (5.13)$$

and write (5.11) as follows:

$$H_{k+1} = \operatorname*{argmin}_J \|JA - C\|_F^2. \quad (5.14)$$

The analytical solution of (5.14) is provided by the normal equations, and the quasi-Newton step (5.5) is

$$p_{k+1} = -H_{k+1} F(x_{k+1}) = -(CA^T)(AA^T)^{-1} F(x_{k+1}). \quad (5.15)$$

It can be obtained from the following procedure:

$$\text{Solve } z = \underset{y}{\operatorname{argmin}} \|Ay - F(x_{k+1})\|_2^2 \tag{5.16}$$

$$\text{Define } p_{k+1} = -Cz \tag{5.17}$$

Indeed, the solution of (5.16) is

$$z = (A^T A)^{-1} A^T F(x_{k+1}) \tag{5.18}$$

and applying (5.18) in (5.17) we obtain

$$p_k = -C(A^T A)^{-1} A^T F(x_{k+1}) \tag{5.19}$$

which is exactly (5.15), pointing out that $A^T(AA^T)^{-1} = (A^T A)^{-1}A^T$. The matrix $(A^T A)^{-1}A^T$ is often called the pseudo-inverse of A (Golub and Van Loan, 1996).

5.3.1 Notes

- The least-squares problem (5.16) is a sparse vector least squares, contrarily to (5.11) that is a generalized matrix least squares. The matrix $Y_{k+1}\Omega$ is dense with a maximum of $n \times \kappa$ entries, and Γ is diagonal with n entries. Therefore, the proportion of non zero entries in A is $1 + \kappa/n + \kappa$. This value decreases when n increases. Also, it can be controlled by the user-defined parameter κ.

- The step p_{k+1} is obtained directly, without any need for the matrix H_{k+1}. This is a key feature of our approach for large-scale problems, as no storage requirement is anymore needed.

- Problem (5.16) can be efficiently solved with a Krylov-based method. In that case, we can avoid storing matrices A and C, and the method can be implemented as a matrix-free algorithm. The only matrices that we need to form are Y_k, S_k, $\widehat{H}^0_{k+1}$, and Γ. This is discussed in section 5.4.

- The analytical solution of (5.11) is given by the normal equations, that is

$$H_{k+1} = \widehat{H}^0_{k+1} + \left(\Gamma^2 + Y_{k+1}\Omega^2 Y_{k+1}^T\right)^{-1} Y_{k+1}^T \Omega^2 \left(S_{k+1} - \widehat{H}^0_{k+1} Y_{k+1}\right). \tag{5.20}$$

  Just as for GSM, the choice $\widehat{H}^0_{k+1} = H_k$ in (5.11) leads to an update formula. In this context we can prove the convergence of the method. The proof is very similar to the one given in section 4.4 as described by Broyden (1965) and therefore as been omitted.

5.4 The algorithm

In this section, we formally describe the algorithm designed to solve large-scale problems, based on the ideas described in Section 5.3. This algorithm has been named iGSM, because it is a kind of "inverse" version of the GSM method proposed in Chapter 4.

Given a starting point x_0, an initial matrix H_0 and κ the maximum size of the population, iGSM is described by:

1. Compute $p_1 = H_0^{-1}F(x_0)$, $k = 1$
2. While no stopping criterion is verified
   (a) $x_{k+1} = x_k + p_k$ and evaluate $F(x_{k+1})$,
   (b) $s_i = x_{k+1} - x_i$ and $y_i = F(x_{k+1}) - F(x_i)$ for $i = m, m+1, \ldots, k-1, k$, $S = (s_k\ s_{k-1}\ \ldots s_m)$ and $Y = (y_k\ y_{k-1}\ \ldots y_m)$ with $m = \max(0, k - \kappa + 2)$,
   (c) $z = \operatorname{argmin}_y \|Ay - F(x_{k+1})\|_2^2$ with A defined by (5.12);
   (d) $p_{k+1} = -Cz$ with C defined by (5.13).
   (e) k=k+1

The critical steps for large-scale applications are Steps 2c and 2d. The least-squares problem in Step 2c must be solved with an iterative Krylov-based method, involving only matrix-vector products. We refer to those iterations as *inner* iterations. We propose to use LSQR (Paige and Saunders, 1982), already described in Section 3.3.2. This method is analytically equivalent to conjugate gradient methods, and exhibits better numerical properties. In particular LSQR is more reliable when the problem is ill-conditioned. Note also that the least-squares problem defined in (2c) is underdetermined and has infinitely many solutions. But LSQR computes the minimum norm solution of (5.16), which is exactly the solution given by (5.18) (Golub and Van Loan, 1996). The main advantage here is that the matrix A is not explicitly needed. Only matrix-vector products of the type Ay and A^Tz are used. With LSQR, as any Krylov-based method, the theoretical number of inner iterations is the size of y, that is $n + \kappa$. If iGSM was implemented directly with the generalized matrix least squares (5.11), the number of iterations would have been $n^2 + n\kappa$. It shows that the procedure (5.16)–(5.17) is a critical feature of our approach. Indeed, when n is large, it affects both the memory requirements and the theoretical upper bound on the number of inner iterations. Furthermore in practice, as illustrated by section 5.5.4, LSQR needs a much smaller number of iterations to find the minimum norm solution of Step 2c.

In Step 2d, matrix C is involved only in a matrix-vector and, again, does not need to be explicitly generated either. To obtain an efficient implementation of Ay, A^Tz and Cz, the size of the population κ and the matrices Γ, Ω and $\widehat{H}_{k+1}^0$ in (5.12) and (5.13) must be carefully defined. This is discussed below.

5.4.1 Implementation notes

- The only dense parts in A and C come from Y_{k+1} and S_{k+1} of size $n \times \tilde{m}$ where $\tilde{m} = \min(k, \kappa)$ with κ the maximum number of previous iterates involved in the computation of the next iterate. For large-scale problems, it is good practice to keep the value of κ low. We have found $\kappa = 10$ to be appropriate for most cases, except when the problem is very large (more than 100,000 variables, say), we preferred then a smaller κ to reduce the size of Y_{k+1} and S_{k+1}. We give in Section (5.5.3) a numerical analysis of the impact of κ on the behavior of the method.

- The weights ω_{k+1}^i for $i = m, \ldots, k$, where $m = \max(0, k - \kappa + 2)$, must reflect how the linear model is supposed to fit the nonlinear function at each point in the population. From Taylor's theorem, we have

$$x_i - x_{k+1} = H_{k+1}(F(x_{k+1}) - F(x_i)) + o(\|F(x_{k+1}) - F(x_i)\|^2). \qquad (5.21)$$

  Therefore, we simply propose

$$\omega_{k+1}^i = \frac{1}{\|F(x_{k+1}) - F(x_i)\|^2} \qquad \forall i = m, \ldots, k. \qquad (5.22)$$

- The role of Γ can be interpreted as a way to guarantee a safe positive definiteness of $\Gamma^2 + Y_{k+1}\Omega^2 Y_{k+1}^T$ in (5.20). Clearly, $Y_{k+1}\Omega^2 Y_{k+1}^T$ is theoretically positive semidefinite. Consequently the only dubious case is when some eigenvalues are numerically close to zero. This is avoided by choosing $\Gamma = \tau I_{n \times n}$ with $\tau \in \mathbb{R}$. Ideally, τ must be as small as possible in order not to perturb the model and, meanwhile, significantly different from zero so that $\Gamma^2 + Y_{k+1}\Omega^2 Y_{k+1}^T$ is numerically invertible. A typical value is $\tau = \sqrt{\epsilon}$ where ϵ is the machine epsilon. An analysis of the impact of τ is presented at Section 5.5.4.

- For $\widehat{H}_{k+1}^0$, we use by default $I_{n \times n}$. We did investigate more sophisticated choices, involving other diagonal matrices, or even bi- or tri-diagonal matrices. But apparently $\widehat{H}_{k+1}^0 = I_{n \times n}$ seems to be the best compromise between efficiency and robustness in this framework.

- The LSQR algorithm has been implemented using the description given by Paige and Saunders (1982). This method includes three different stopping criteria defined by the three dimensionless quantities, following the notation of Paige and Saunders (1982), we have ATOL=BTOL=10^{-16} and CONLIM=10^{10}.

5.5 Numerical results

In this section we present numerical results using the proposed algorithm described in section 5.4. Before analyzing large-scale problems, we first expose a

performance analysis of iGSM method to solve medium-scale systems of nonlinear equations, extending the results presented in chapter 4. Then we show, on a set of classical difficult problems, the efficiency of iGSM on large-scale systems. We also illustrate the behavior of iGSM on specific problems to underline the influence of various parameters, such as the size of the population and the value of τ, on the performance of the algorithm. Finally we propose some results on noisy systems of nonlinear equations to underline the robustness of our population approach compared to classical large-scale methods in presence of different types of stochasticity. In Chapter 6, we illustrate the applicability of iGSM to the consistent anticipatory route guidance generation problem, a noisy fixed point problem of high dimension.

5.5.1 Validation of the adaptations

In order to illustrate the impact, in terms of performance, of the adaptation granted to GSM to handle large-scale problems, we propose to show the performance of iGSM compared to GSM. GSM is implemented with the parameters given in section 4.5 and iGSM with the features described at the end of Section 5.4. The numerical experiments have been carried out on the same set of test functions as in Section 4.5.1, with dimensions 10 and 50, and using the identity matrix as the initial approximation of the Jacobian. The results are presented using the performance profiles proposed by Dolan and More (2002). It is a powerful visual tool for evaluating and comparing the performance of several algorithms applied to many problems, it is described in Section 4.5.3. The performance profile for a method is the cumulative distribution function for a given performance metric. In the following we use the number of function evaluations to reach convergence as performance metric. The results considered here include all problems, not only problems which have been solved at least by one of the two considered methods

The performance profile presented in Figure 5.1 shows that iGSM (with large-scale features) has a similar behavior as GSM. Actually, GSM performs a little faster (small values of π) than iGSM. GSM (resp. iGSM) is the faster method on nearly 45% (resp. less than 40%) of the problems. Astoundingly iGSM seems a little more robust than GSM, solving about 70% (high values of π), but we do not consider these differences being significant. This is encouraging, as the large-scale features are not associated with a loss of performance, compared to the original GSM design.

5.5.2 General performance analysis

The numerical experiments for the large-scale problems were carried out using a set of 33 problems. The first 18 problems are classical systems of nonlinear equations.

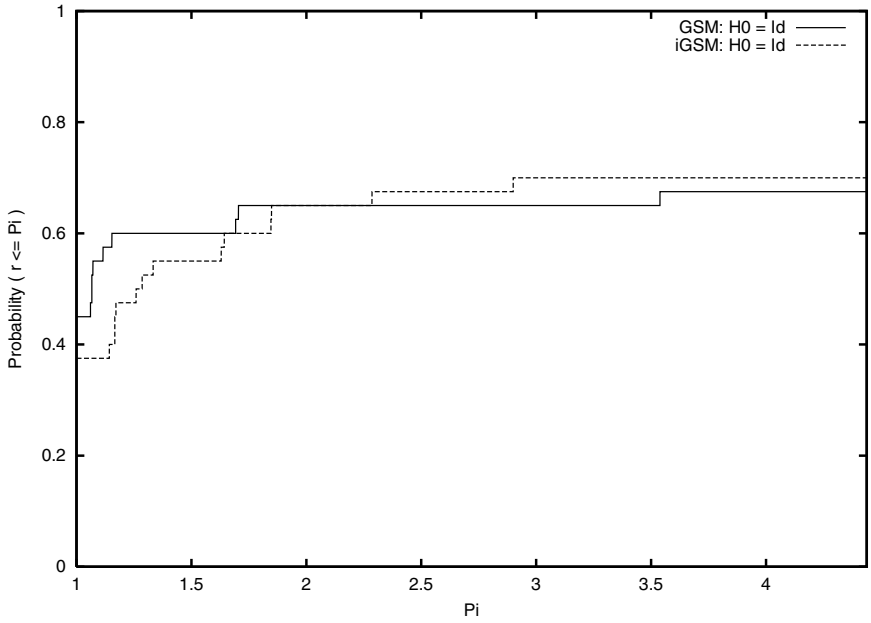


Figure 5.1: Performance profiles for medium scale problems

- Bogle and Perkins, 1990:
  - Countercurrent reactors problem [CRP],
- Toint, 1986
  - A trigonometric system [TS],
  - A trigonometric-exponential system (I),
- Gomes-Ruggiero et al., 1991
  - Singular Broyden problem [SBP],
  - Structured Jacobian problem [SJP],
- Li, 1989
  - Tridiagonal System [TdS],
  - Five-diagonal system [FdS],
  - Seven-diagonal system [SdS],
- Moré et al., 1981

  - Extended Powell badly scaled function [EPBSF],
  - Broyden tridiagonal function [BTF],
  - Broyden Banded problem [BBP],
  - Boundary value problem [DBVP],

- Kelley, 2002b
  - Chandrasekhar H-equation residual [CHR],
  - Ornstein-Zernike Equation [OZE],
  - Convection-Diffusion Equation [CDE],

The remaining 15 systems without name have been selected from Spedicato and Huang (1997). Their number coincides with the numbering used in the paper.

- Spedicato and Huang, 1997: [SPED1], [SPED2], [SPED4], [SPED5], [SPED6], [SPED7], [SPED9], [SPED12], [SPED13], [SPED17], [SPED18], [SPED20], [SPED22],

- Roose et al., 1990: [SPED22],

- Robert and Shipman, 1976: [SPED27],

- Ascher and Russel, 1985: [SPED28].

For most problems, the size could be easily adjusted. For the tests we have generated an instance with $n = 100$ and an instance with $n = 1000$. For the Ornstein-Zernike Equation, we have chosen $n = 402$, and $n = 961$ for the right preconditioned Convection-Diffusion Equation. For each problem, we have used the starting point proposed in the original papers. We have implemented all test functions with the package Octave (Eaton, 1997). For [CHR], [OZE] and [CDE], we have used the implementation by Kelley (2002b). All tests have been performed in double precision on a 1066MHz Pentium-V running Linux. The machine epsilon is about 2.2204e-16.

We have first compared the three following algorithms.

1. Our implementation of iGSM, with the features described at the end of Section 5.4.
2. Our implementation of ICUM, based on Martinez and Zambaldi (1992).
3. Kelley's matrix-free implementation of the GMRES method (Kelley, 2002b).

Note that iGSM and ICUM are not globally convergent, while GMRES is. For iGSM and ICUM, it is good practice to initialize the method with a good approximation of the true Jacobian matrix, estimated by finite differences. In a large-scale context, and when the functions under consideration are expensive to evaluate, there is a trade-off between the resources to compute H_0 and the impact of the initialization on the methods efficiency. In addition to the costless $H_0 = I_{n\times n}$, we have tested four different banded approximations of the Jacobian, that is a 1-diagonal, a 3-diagonal, a 5-diagonal and a 7-diagonal matrix. Note that a p-diagonal finite differences approximation of the Jacobian can be evaluated with p function evaluations (Kelley, 2002b).

We have used the classical stopping criterion proposed by (Dennis and Schnabel, 1996):

$$\|F(x_k)\| \leq 10^{-6}\|F(x_0)\| \tag{5.23}$$

We also stop an algorithm after a computation budget of 200 function evaluations, or when $\|F(x)\|_2 > 10^{10}$. In these cases, we consider that it has failed to converge.

Table 5.1 gives the detailed results, where each row corresponds to a problem and each column to an algorithm. For iGSM and ICUM, the choice of H_0 is specified by "Id" if H_0 is the identity, and by n-d if H_0 is a n-banded approximation of the Jacobian matrix. Each entry contains the total number of function evaluations (including those used to evaluate H_0), if the algorithm has converged. If the computation budget has been reached before convergence, it contains an "m". If the algorithm has stopped because $\|F(x)\|_2 > 10^{10}$, there is an "x". For GMRES, "a" means that the algorithm has not completed the parabolic line search after 40 step reductions.

This general results are presented using the performance profiles described in Section 4.5.3, using the number of function evaluations to reach convergence as a performance metric.

Full performance profiles

Figure 5.2 gives the performance profiles of all methods run on all problems. We observe that ICUM using the identity matrix as initial approximation of the Jacobian is by far the worst method in term of robustness, but interestingly when it converges it is the fastest algorithm characterized by its horizontal profile. We also note that the most robust method is iGSM using a 7-diagonal approximation of the Jacobian as starting matrix. However, due to the high number of variants, this profile is difficult to decrypt. Moreover, GMRES is competing against 5 versions of iGSM and 5 versions of ICUM, which is not completely fair. Therefore, we start by analyzing the variants of iGSM and ICUM, and select one of them for further comparisons.

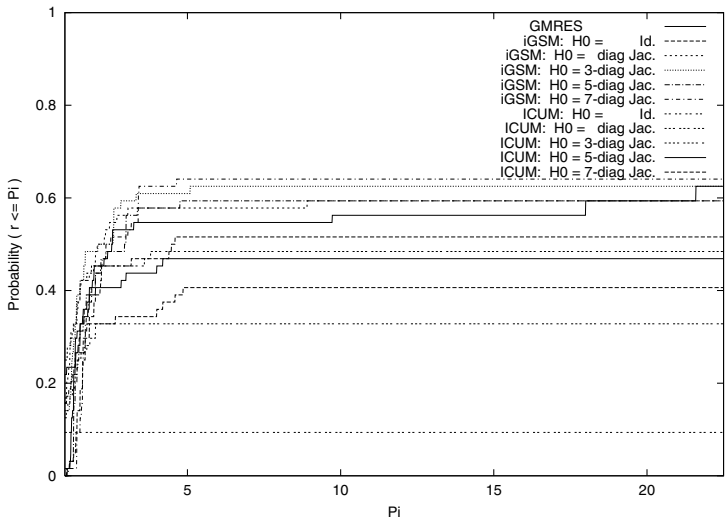


Figure 5.2: General performance profiles

iGSM - initial H_0

Figure 5.3 plots the profile of each instance of iGSM. It clearly emphasizes that $H_0 = I_{n\times n}$ is a poor choice in terms of robustness for the method. As anticipated the best approximation (7-diagonal) produces the most robust method as it solves nearly 65% of the problems within our computational budget. But choosing a diagonal or a 3-diagonal approximation of the inverse of the Jacobian seems to be a better compromise between efficiency and reliability. For higher order approximation (5 or 7-diagonal approximation) the efficiency is penalized by the number of function evaluations necessary to compute the initial Jacobian.

ICUM - initial H_0

We have performed a similar analysis for ICUM. Note the flat slope of the profile of ICUM considering $B_0 = I_{n\times n}$. Additionally the robustness (high value of π in Figure 5.4) of the different instances of ICUM follows the quality of the initial approximation matrix. This is consistent with Luksan and Vlcek (1998), who reports that ICUM method is efficient when it has a good approximation of the starting Jacobian, but lacks of robustness when the approximation in not precise.

In the following, we have selected iGSM with a diagonal approximation and ICUM with a 3-diagonal approximation of the inverse of the Jacobian.

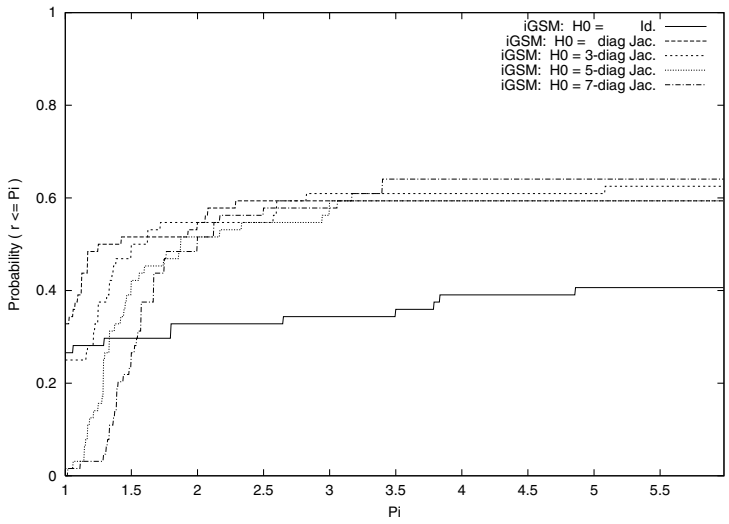


Figure 5.3: Performance profiles of iGSM with different B_0

Performance analysis

The comparison of iGSM 1-d, ICUM 3-d and GMRES is provided in Figure 5.5. It is interesting to note that the three methods are almost equivalent in terms of efficiency, as each of them wins about 30% of the time. However, the robustness of ICUM is significantly under the others. The curve for iGSM and GMRES are very similar. Comparing the methods pairwise confirms the similar behavior of iGSM and GMRES (Figure 5.6(a)). The superiority of iGSM over ICUM, both in efficiency and robustness, is clearly depicted in Figure 5.6(b). iGSM wins about 50% of the times, and solves about 60 % of the problems, while ICUM wins about 30% of the times, and solves about 50 % of the problems. Remark also the small range for values of π in Figure 5.6(b) illustrating that, when one of the two methods converges, the other converges with less than 3.5 times the number of function evaluations. Finally, we point out that the average number of inner iterations of the LSQR algorithm (reported in Table 5.2) is significantly lower than the size of the problem.

These results are very encouraging. In the quasi-Newton literature, ICUM is considered as one of the best secant methods solving large-scale systems of nonlinear equations (Spedicato and Huang, 1997, Ruggiero et al., 1996, Luksan and Vlcek, 1998 and Martinez, 2000). The significant improvement brought by iGSM is noticeable. Moreover, Newton-Krylov methods are recognized as the most efficient methods to date to solve large-scale problems (Kelley, 2002b). Therefore, the fact that our method reaches the same level of performance is an achievement, taking into consideration that the implementation of GMRES involves globalization techniques which is not the case for iGSM.

We note that the experiments presented here involve functions that are

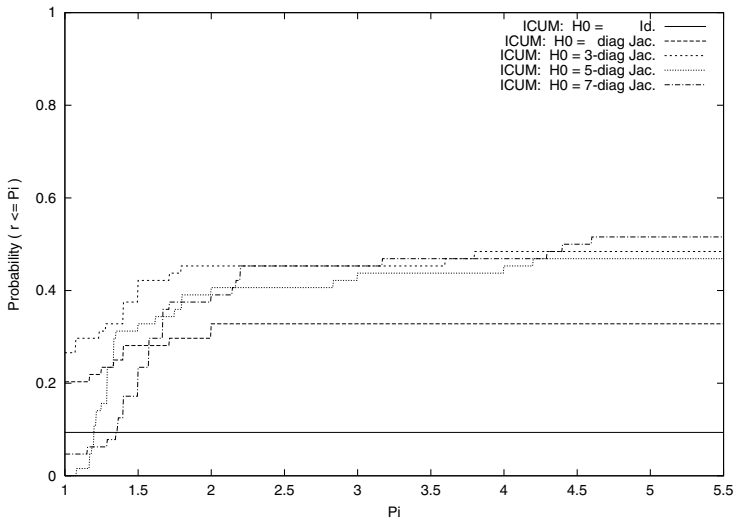


Figure 5.4: Performance profiles of ICUM with different B_0

cheap to evaluate. In this case, our method exhibits an overhead for the management of the population and the solution of the least-squares problem. Moreover, the current Octave implementation is certainly not optimized. It contains several checking procedures, and is based on simple data structures. It may certainly be improved. Fortunately, even in such an unfavorable context, the time overhead is not important, as illustrated by Figure 5.7, where the performance profiles for the actual running times of the algorithms are reported (see Table 5.2 for actual numbers). The overhead appears only for small values of π, and the profile increases rapidly. Anyway, this overhead disappears when the dominant computation cost is the function evaluation.

In the rest of the section, we will analyze more closely the behavior of the algorithm on the Convection-Diffusion problem described by Kelley (2002b). It is a finite differences discretization of second order nonlinear boundary value problem. The function is right preconditioned using a fast Poisson solver. The size of the problem is 961. We use the implementation furnished by Kelley (2002b). This problem belongs to an important class of large-scale problems, usually discussed in the literature (Martinez and Zambaldi, 1992, Luksan and Vlcek, 1998, Ruggiero et al., 1996 and Kelley, 2002b).

Figure 5.8 reports the evolution of the relative residual $\|F(x_k)\|/\|F(x_0)\|$ with the number of function evaluations for iGSM and GMRES. Note that ICUM does not converge on this problem. Clearly, iGSM converges faster than GMRES. Compare the smooth curve for iGSM with the horizontal "steps" of the GMRES curve. Those steps correspond to stages of the algorithm when partial derivatives of F are evaluated by finite differences.

We analyze the impact of the two important parameters of the algorithm

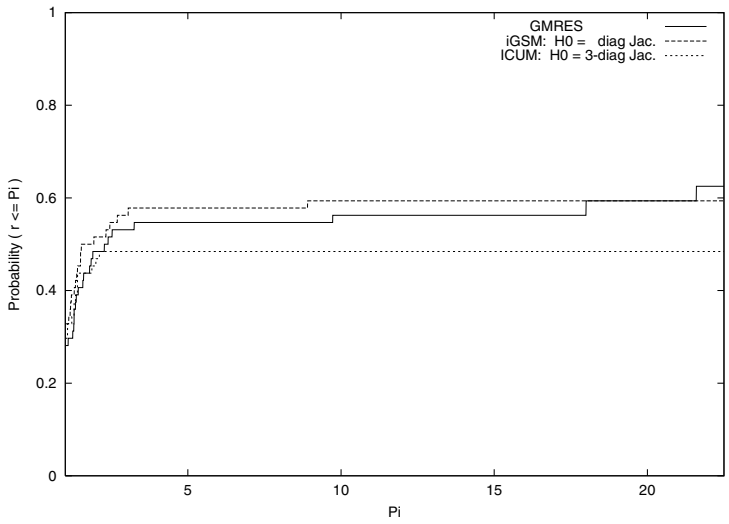
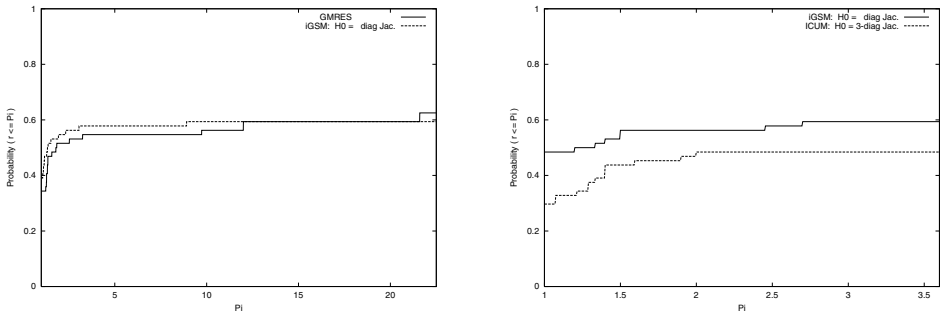


Figure 5.5: Performance Profiles for iGSM 1-d, ICUM 3-d and GMRES

on this problem. First, the impact of the size κ of the population is described in Section 5.5.3. The impact of parameter τ is described in Section 5.5.4.

5.5.3 Impact of the size of the population

We analyze the impact of the parameter κ, the maximum size of the population, on the behavior of iGSM. Table 5.3 reports, for each size of the population, the number of function evaluations and the time to reach convergence. When the number of function evaluations is denoted by m, it means that the algorithm has reached the computational budget. With a population of two iterates, the method does not converge. Clearly, using more than two iterates signif-



(a) GMRES and iGSM

(b) ICUM and iGSM

Figure 5.6: Head-to-Head Performance Profiles

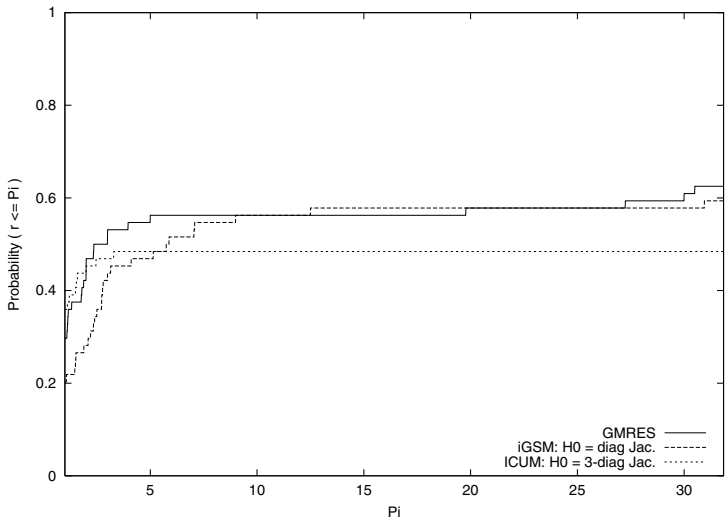


Figure 5.7: Performance Profiles (Time) for iGSM 1-d, ICUM 3-d and GMRES

icantly improves the algorithm performance, illustrating the added value of our population-based approach contrasting with other classical quasi-Newton methods. More interestingly, the performance improvement reaches a plateau around a population of 10. This is good news for large-scale applications, because the size of matrix A defined by (5.12) remains small. We have observed that a population of 10 is appropriate for most problems we have tested.

5.5.4 Impact of the parameter τ

We also illustrate the influence of the choice of the parameter τ on the efficiency of the algorithm. Table 5.4 reports (i) the number of evaluations of the function, (ii) the time of computation necessary to reach convergence, and (iii) the average number of inner iterations of the LSQR algorithm for different values of τ. It shows that if τ is too big (that is 1, 10^{-1} and 10^{-2}), too much weight is assigned to the a priori $\widehat{H}^0_{k+1}$ and the method fails to converge. For smaller values of τ the method converges, but consumes in average more inner iterations of LSQR when the parameter becomes smaller. Indeed, with small values of τ, the least-squares problem is more likely to be ill-conditioned. For $\tau \in [10^{-11}, 10^{-4}]$ its influence is not fundamental with regard to the number of function evaluations needed by iGSM to converge, it only affects the average number of inner iterations of LSQR. We have observed that $\tau = \sqrt{\epsilon}$ where ϵ is equal to the machine epsilon is appropriate for most problems we have tested.

Figure 5.9 gives the details of the number of inner iterations used by LSQR, on the Convection-Diffusion equation, to reach convergence at each main iteration of iGSM (implemented with the choice of parameters defined in Sec-

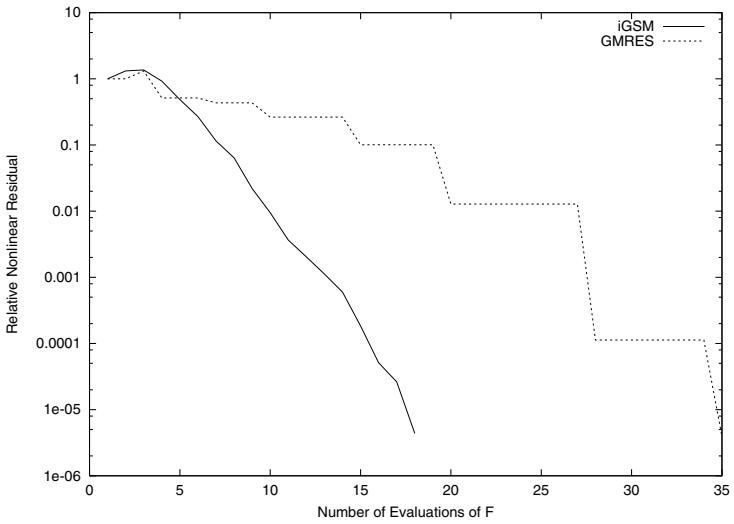


Figure 5.8: Convection-Diffusion equation

tion 5.4.1). The first main iteration does not use LSQR. In the early iterations of the algorithm, the size of the population is low and the associated least-squares problem is simple to solve, explaining the small number of inner iterations used by iGSM. As more iterates are added to the population, the complexity of the least-squares problem increases. However, the number of inner iterations stays well below the theoretical upper bound $n + \kappa$, which is 971 in this case.

5.6 Behavior in presence of noise

Solving systems of equations involving data collection processes or simulator runs may be intricate due to the stochastic nature of the underlying system. Therefore, we present an analysis of the behavior of our method in the presence of noise.

As in Section 4.5.5, following Choi and Kelley (2000), we consider a smooth deterministic system of nonlinear equations $F_s(x) = 0$ perturbated by a small mapping $\phi(x)$, that defines a noisy system of nonlinear equations $G(x) = 0$ as follows:

$$G(x) = F_s(x) + \phi(x) \tag{5.24}$$

where $\phi(x)$ is a white noise such that its variance decreases in the vicinity of the solution, that is

$$\phi(x) \sim N(0, \alpha^2 \|x - x^*\|^2), \tag{5.25}$$

where $\alpha \in \mathbb{R}$.

To perform the analysis, we have chosen a problem where ICUM, GMRES and iGSM have a similar performance in the case without noise, in order to

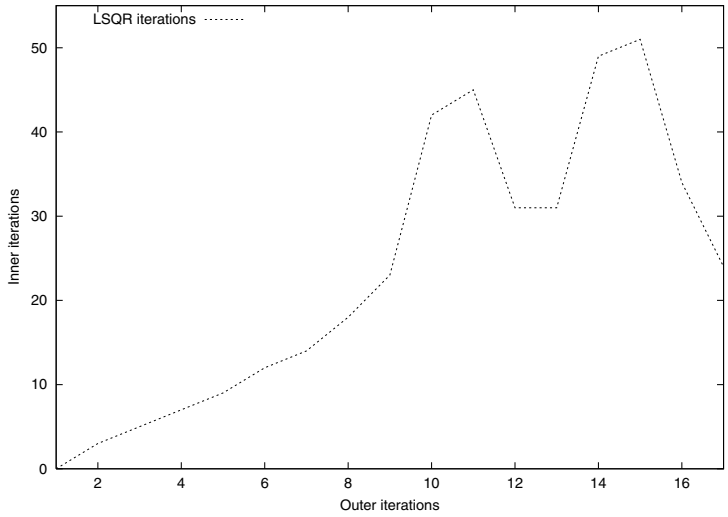


Figure 5.9: Inner LSQR iteration

clearly isolate the effect of the noise on the algorithm performance. We have selected problem [SPED12]. It is defined as

$$\begin{cases} f_1 = x_1, \\ f_i = \cos(x_{i-1}) + x_i - 1, \quad \forall i = 1, \ldots, n \end{cases} \tag{5.26}$$

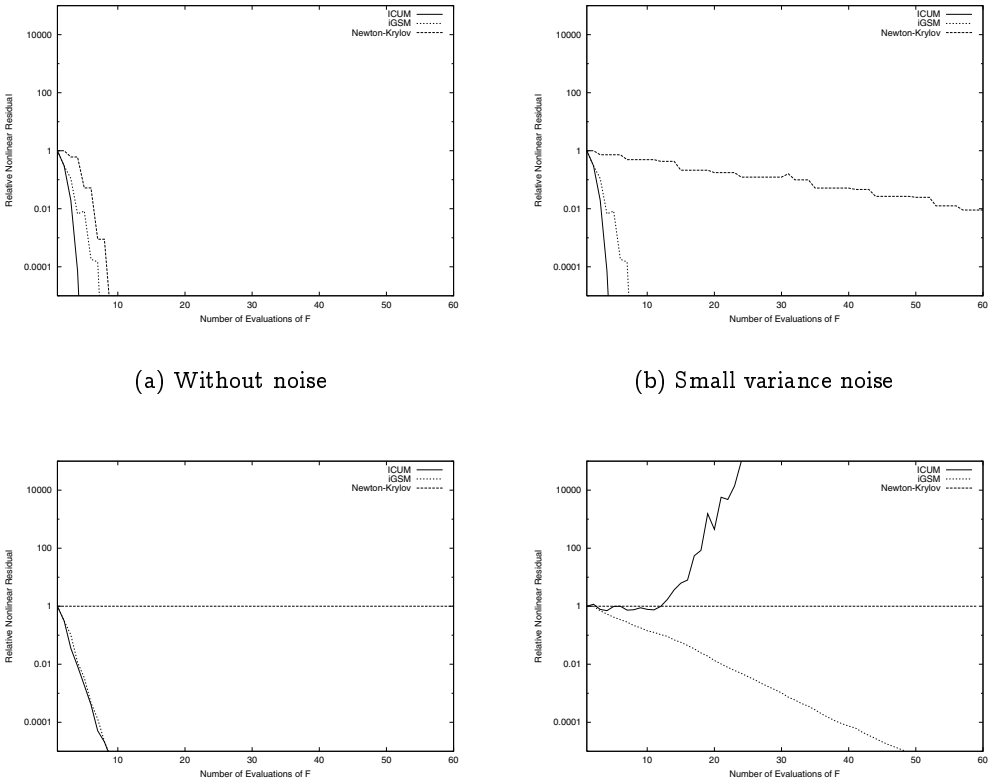
with initial point $x_0 = (0, \ldots, 0)$. Figure 5.10 reports the relative nonlinear residual of the smooth system, i.e. $\|F_s(x)\|/\|F_s(x_0)\|$ as a function of the number of evaluations of F.

In the presence of a very small noise ($\alpha = 10^{-8}$, Figure 5.10(b)), we observe a significant decrease in the performance of the Newton-Krylov method, compared to the two others which are not notably affected. This makes sense, as Newton-Krylov relies on finite differences approximations of the derivatives, which are sensitive to the noise. When the magnitude of the noise increases ($\alpha = 10^{-2}$ (Figure 5.10(c)), the GMRES method is stalled at the starting point. The ICUM method is slightly disrupted, while iGSM still converges in 8 iterations. For larger noise ($\alpha = 10^{-1}$, Figure 5.10(d)), the ICUM method does not make much progress in the early iterations, and starts diverging after about 12 function evaluations. iGSM still converges but uses more function evaluations, about 47.

We have also considered problems with a constant noise, that is where

$$\phi(x) \sim N(0, \alpha^2). \tag{5.27}$$

Again, Newton-Krylov is already affected by a small noise ($\alpha = 10^{-8}$, Figure 5.11(b)). For the medium noise ($\alpha = 10^{-2}$, Figure 5.11(c)) and the large noise ($\alpha = 10^{-1}$, Figure 5.11(d)), we observe that Newton-Krylov is stalled,



(a) Without noise

(b) Small variance noise

(c) Medium variance noise

(d) Big variance noise

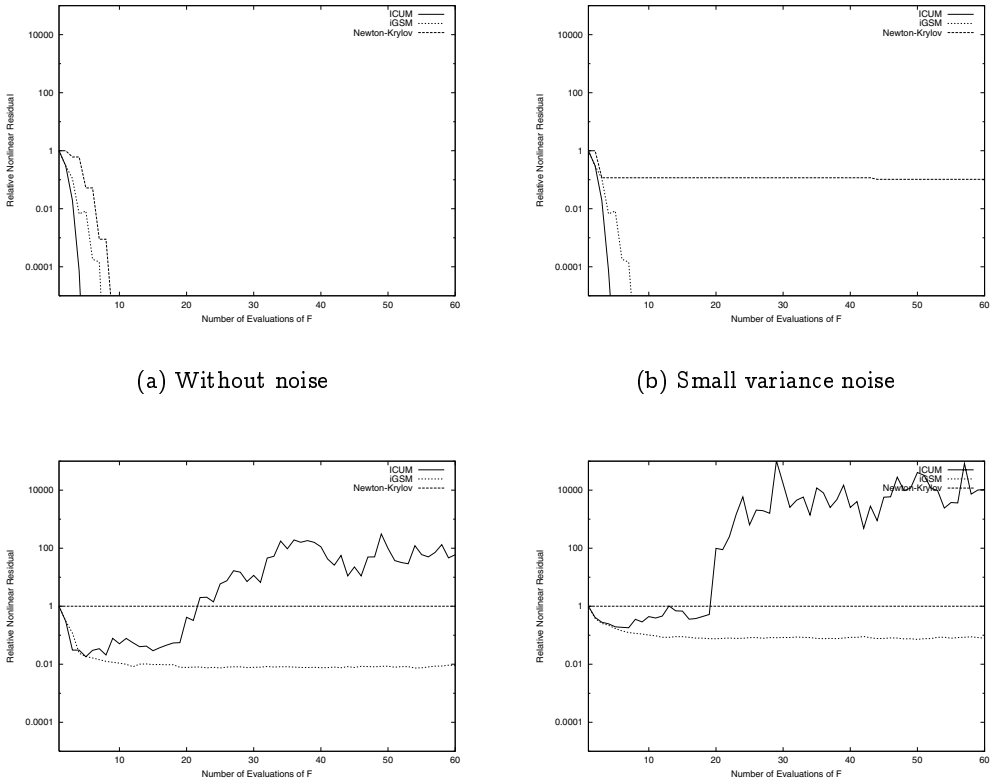
Figure 5.10: Behavior with stochasticity

ICUM decreases first and then explodes, and iGSM decreases to reach the noise level.

This preliminary analysis emphasizes the added value of the iGSM method in the presence of noise, when GMRES cannot be used. In Chapter 6, we show that iGSM works on a very large-scale noisy problem, that GMRES and ICUM cannot solve.

5.7 Conclusion

We have proposed an adaptation of the generalized secant method described in chapter 4 to solve large-scale systems of nonlinear equations. Because GSM algorithms are not intended to deal with large-scale problems we have designed a new algorithm based on the same idea considering a population of previous iterates to calibrate the linear model. This calibration is based on a least-squares approach instead of exact interpolation methods. Combining this approach with the update, at each iteration, of the inverse of the approximation of the Jacobian leads to a powerful algorithm performing only matrix-vector products,



(a) Without noise

(b) Small variance noise

(c) Medium variance noise

(d) Big variance noise

Figure 5.11: Behavior with stochasticity

making it very cheap in terms of computation time and memory consumption.

In order to validate the intended adaptations we have shown its performance on standard systems of nonlinear equations and compared the results to classical algorithms and GSM. These results are very impressive on medium-scale problems, as the efficiency of iGSM is very similar to GSM algorithms and clearly outperforms standard secant methods, and especially ICUM currently considered as one of the best method solving large-scale systems of nonlinear equations without derivatives. Tests on large-scale problems are very convincing as iGSM is even able to compete with Newton-Krylov methods, which use derivatives information using finite differences. Moreover in terms of reliability, in presence of a computation budget, these two types of methods have a similar behavior. As iGSM uses a population of previous iterates to calibrate the approximation of the Jacobian, we demonstrate its ability to handle stochasticity. Even though Newton-Krylov methods are useless in presence of noise due to their use of finite differences and ICUM is sensitive to stochasticity, as it calibrates the approximation of the Jacobian matrix using only the last two iterates. Moreover in the next chapter we present the application of iGSM to

solve the consistent anticipatory route guidance problem. We solve this problem on a real network to demonstrate the applicability of our algorithm to very large problems.

The results obtained with iGSM indicates that this method will be one of the first choices to solve large-scale systems of nonlinear equations, especially in presence of noise. We plan to investigate globalization to this algorithm but as pointed out by Friedlander et al. (1997):"purely monotone strategies can reject iterates belonging to rapidly convergent sequences, even if they are relatively close to the solution. Moreover, strictly monotone strategies usually have local-nonglobal minimizer of $\|F(x)\|$ as attraction point, which are completely useless when our aim is to solve $F(x) = 0$." They propose nonmonotone strategies to alleviate these kind of problems. We think that the next step will be to globalize iGSM using this kind of nonmonotone strategies hoping an increase of performances in terms of robustness.

Finally, as proposed for GSM, the ideas described in this chapter to handle large-scale systems of nonlinear equations will have to be considered in the context of optimization.

		GMRES	iGSM					ICUM				
Problem	Size		Id	1-d	3-d	5-d	7-d	Id	1-d	3-d	5-d	7-d
SPED1	100	m	x	x	x	x	x	x	x	x	x	x
	1000	m	x	x	x	x	x	x	x	x	x	x
SPED2	100	9	6	7	9	14	13	6	8	9	18	13
	1000	9	6	7	9	13	15	6	7	9	17	19
SPED4	100	108	9	5	13	15	17	x	5	7	9	11
	1000	108	9	5	13	15	17	x	5	7	9	11
SPED5	100	21	x	32	m	m	x	x	x	m	m	m
	1000	23	x	27	x	x	x	x	24	43	x	103
SPED6	100	m	m	x	x	x	x	x	x	x	x	x
	1000	m	m	x	x	x	x	x	x	x	x	x
SPED7	100	37	32	x	42	44	46	x	x	26	28	30
	1000	35	31	x	42	44	46	x	x	32	42	26
SPED9	100	a	x	m	x	x	x	x	x	x	x	x
	1000	a	x	m	m	m	m	x	x	x	x	x
SPED12	100	9	8	10	13	15	17	5	10	19	21	23
	1000	9	8	9	13	15	17	5	10	18	20	22
SPED13	100	m	m	x	6	8	10	m	x	6	8	10
	1000	m	m	x	6	8	10	m	x	6	8	10
SPED17	100	m	x	x	x	x	x	x	x	x	x	x
	1000	m	x	x	x	x	x	x	x	x	x	x
SPED18	100	m	m	x	7	9	11	m	x	7	9	11
	1000	m	m	x	7	9	11	m	x	7	9	11
SPED20	100	m	73	m	m	m	m	m	m	m	m	m
	1000	m	76	72	m	m	m	m	m	m	m	m
SPED22	100	19	42	14	12	14	16	x	14	10	12	14
	1000	16	46	12	12	14	16	x	14	10	12	14
SPED27	100	m	m	x	5	8	10	m	x	6	7	9
	1000	m	m	x	6	8	10	m	x	6	7	9
SPED28	100	m	m	x	11	13	15	m	x	8	10	12
	1000	m	m	x	8	10	11	m	x	7	9	12
CRP	100	m	x	180	x	m	m	x	x	x	x	x
	1000	m	x	m	x	x	x	x	x	x	x	x
EPBSF	100	m	36	37	36	38	40	x	x	x	x	x
	1000	m	26	37	36	38	40	x	x	x	x	x
TS	100	39	m	12	61	m	38	m	12	x	m	m
	1000	107	m	11	m	m	m	m	12	m	m	m
TESI	100	19	x	22	x	x	x	x	29	x	x	x
	1000	16	x	22	x	x	x	x	29	x	x	x
SBP	100	28	x	m	m	m	m	x	x	x	x	x
	1000	46	x	89	153	155	157	x	46	x	x	x
TdS	100	43	x	m	m	m	m	x	x	x	x	x
	1000	22	x	196	m	m	m	x	x	x	x	x
FdS	100	55	x	71	m	m	m	x	x	x	x	x
	1000	52	x	121	m	113	115	x	x	x	x	x
SdS	100	64	x	m	133	m	m	x	x	x	x	x
	1000	37	x	113	m	176	172	x	x	x	x	x
SJP	100	19	x	14	17	18	21	x	14	15	17	19
	1000	22	68	14	17	16	21	x	14	15	17	18
ERF	100	108	8	9	10	12	14	x	x	6	8	10
	1000	108	8	9	10	12	14	x	x	6	8	10
EPSF	100	46	18	37	24	x	25	x	x	x	x	35
	1000	46	18	36	24	x	25	x	x	x	x	35
EGLF	100	43	22	17	30	32	34	x	29	x	x	x
	1000	43	22	17	48	50	52	x	x	x	x	x
BTF	100	25	x	27	14	16	18	x	x	11	13	15
	1000	24	x	27	13	15	17	x	x	10	12	14
BBP	100	20	53	20	25	26	28	x	20	28	27	27
	1000	19	72	19	22	23	25	x	18	23	27	27
DBVP	100	m	m	x	7	9	11	m	x	7	9	11
	1000	m	m	x	7	9	11	m	x	7	9	11
CHR	100	12	9	9	11	13	15	8	10	12	14	16
	1000	12	7	16	18	13	14	7	12	12	14	15
OZE	402	79	x	m	m	m	m	x	x	x	x	x
CDE	961	35	17	18	20	22	25	x	x	x	x	x

Table 5.1: General Results: number of function evaluations

		GMRES		iGSM			ICUM	
Problem	Size	Eval.	Time [sec]	Eval.	Time	Aver. LSQR iter.	Eval.	Time [sec]
SPED1	100	m	-	x	-	-	x	-
	1000	m	-	x	-	-	x	-
SPED2	100	9	0.3	7	0.1	2	9	0.1
	1000	9	1.2	7	2.5	2	9	1.9
SPED4	100	108	6.1	5	0.2	3	7	0.4
	1000	108	53.4	5	2.9	3	7	2.7
SPED5	100	21	0.4	32	2.3	2	m	-
	1000	23	8.7	27	19.2	2	43	21.4
SPED6	100	m	-	x	-	-	x	-
	1000	m	-	x	-	-	x	-
SPED7	100	37	1	x	-	-	26	0.5
	1000	35	6,7	x	-	-	32	10.3
SPED9	100	a	-	m	-	-	x	-
	1000	a	-	m	-	-	x	-
SPED12	100	9	0.2	10	0.3	13.5	19	0.3
	1000	9	1.4	9	3.8	10.3	18	4.6
SPED13	100	m	-	x	-	-	6	0.1
	1000	m	-	x	-	-	6	1
SPED17	100	m	-	x	-	-	x	-
	1000	m	-	x	-	-	x	-
SPED18	100	m	-	x	-	-	7	0.1
	1000	m	-	x	-	-	7	1.7
SPED20	100	m	-	m	-	-	m	-
	1000	m	-	72	53.6	20.2	m	-
SPED22	100	19	0.5	14	0.9	19.7	10	0.1
	1000	16	3	12	6.3	18.3	10	2.7
SPED26	100	m	-	x	-	-	6	0.1
	1000	m	-	x	-	-	6	1
SPED27	100	m	-	x	-	-	8	0.2
	1000	m	-	x	-	-	7	1.9
CRP	100	m	-	180	27.5	23.8	x	-
	1000	m	-	m	-	-	x	-
EPBSF	100	m	-	37	3.1	3.3	x	-
	1000	m	-	37	29.5	3.7	x	-
TS	100	39	1.4	12	0.7	12.33	x	-
	1000	107	28.5	11	7.2	12.5	m	-
TESI	100	19	1	22	2.8	32.65	x	-
	1000	16	6.9	22	18.8	33.5	x	-
SBP	100	28	0.7	m	-	-	x	-
	1000	46	8.8	89	62	25.1	x	-
TdS	100	43	1.3	m	-	-	x	-
	1000	22	5.6	196	173.3	36.9	x	-
FdS	100	55	2.8	71	11.5	36.5	x	-
	1000	52	21.9	121	112.4	44.5	x	-
SdS	100	64	4	m	-	-	x	-
	1000	37	21.1	113	124	40.3	x	-
SJP	100	19	0.9	14	1.2	23.7	15	0.5
	1000	22	8.7	14	9.7	20.8	15	6.6
ERF	100	108	6	9	0.5	3.3	6	0.2
	1000	108	54.5	9	6.3	3.5	6	2
EPSF	100	46	8.3	37	7.2	25.4	x	-
	1000	46	79.1	36	69.8	7.8	x	-
EGLF	100	43	7.7	17	2.1	7.7	x	-
	1000	43	73.7	17	31.3	6.7	x	-
BTF	100	25	0.6	27	2.5	20.5	11	0.2
	1000	24	4.6	27	18.4	24.7	10	2.6
BBP	100	20	1.7	20	3.2	25.7	28	1.9
	1000	19	15.1	19	22.9	31.8	23	18.2
DBVP	100	m	-	x	-	-	7	0.1
	1000	m	-	x	-	-	7	1.9
CHR	100	12	0.2	9	0.3	11.6	12	0.1
	1000	12	6.6	16	16.6	19.5	12	6
OZE	402	79	1.4	m	-	-	x	-
CDE	961	35	26.4	18	14.1	21.8	x	-

Table 5.2: Time of computation

Population size	Function evaluations	Time [sec]
2	max	-
3	43	30.3
4	38	26.7
5	36	26.2
6	29	21.6
7	21	15.9
8	21	16.2
9	19	14.9
10	18	14.1
11	17	14
12	18	14.9
13	18	15.4
14	18	15.6
15	19	16.8
16	19	16.9
17	19	17.3
18	19	17.4
19	19	17.2
20	19	17.2

Table 5.3: Impact of the size of the population on iGSM

τ	Evaluations	Time [sec]	Average LSQR Iterations
1	x(11)	11.7	2.1
1e-1	x(11)	12.2	2.5
1e-2	x(12)	12.8	2.8
1e-3	21	24.10	16.4
1e-3	18	20	16.9
1e-4	18	20.6	20.9
1e-5	18	20.9	21.8
1e-6	18	21	23.5
1e-7	18	21.1	25.3
1e-8	18	21.3	24.8
1e-9	18	21.2	26.4
1e-10	18	21.4	28.3
1e-11	m	251	14.9
1e-12	m	246.8	11.9
1e-13	m	249.7	15.6
1e-14	m	244.3	12.9
1e-15	m	244.5	14.4
1e-16	m	244.1	13.8

Table 5.4: Parameter τ analysis

Chapter 6

Consistent guidance

Contents

6.1 Introduction

Anticipatory guidance[1] informs travelers about the traffic conditions they will experience during their trip based on prediction of the future conditions of the network. But as traffic information affects drivers behavior, it may invalidate predicted traffic conditions that were used to generate it. Therefore, the concept of consistency is essential for the generation of reliable information. The guidance is called consistent when the forecasts on which it is based are verified after the reaction of drivers. This problem has been formulated as a fixed point problem. Due to the fact that the computation of anticipatory guidance involves complex simulation tools, the distinctive features of this fixed point problem are the presence of stochasticity, the high dimension, and the expensive computation time. Methods for solving the Consistent Anticipatory Route Guidance (CARG) problem are classical fixed point methods. Unfortunately, as pointed out by Bottom et al. (1999), their slow behavior makes them inadequate for real-time applications. In this chapter different formulations of the CARG problem are presented and their associated resolution algorithms are analyzed.

At first Bierlaire and Crittin (2001) have considered the CARG as a large-scale optimization problem without derivative. The idea was to consider a population-based procedure to gather variational information about a local model of the objective function. The preliminary numerical results were encouraging, especially for real-time applications, as the method rapidly decreases the objective function in the first iterations. The main drawback of this kind of algorithms lies in the fact they can get stuck in a local minimizer far from the solution of the original problem. In this chapter the CARG problem is formulated as the resolution of a system of nonlinear equations. Due to the distinctive features of the problem, we propose to use the algorithm iGSM described in Chapter 5 to solve it.

This chapter is organized as follows. In Section 6.2 the analysis framework of the CARG problem, proposed by Bottom (2000), is specified. Section 6.3 describes a number of formulations of this problem associated with their class of resolution algorithms. The numerical results are presented using two software systems designed to produce anticipatory route guidance. In Section 6.4.1 a comparison of performances between fixed point algorithms and iGSM are given using a simulator implemented by Bottom (2000). It allows an analysis of the impact of guidances, with the same level of inconsistency obtained with different algorithms, on the resulting network conditions. In Section 6.4.2 results of a number of simulation experiments are presented using Dynamit on four networks. The Florian network is a small synthetic network composed of 10 links. The Central Artery/Third Harbor Tunnel network is currently in

[1] As specified in the introduction, the terms *guidance* and *information* will be used interchangeably and will refer to any prescriptive and/or descriptive messages given to the drivers.

construction at Boston and composed of 211 links. The Irvine network is a large highways network in California composed of 618 links. Finally the Swiss network is an interurban network on the Swiss plateau composed of 1661 links. These networks allow to compare the results obtained with fixed point methods and iGSM. Large networks also demonstrate the applicability of iGSM on large-scale noisy problems. For example, the Swiss network generates a fixed point problem counting more than 120'000 variables.

6.2 Formulation of the CARG

Route guidance refers to information disseminated to road users with the intent of influencing their route choice decisions. We are interested here in *anticipatory* route guidance where real-time traffic conditions are used to make predictions of the evolution of the network. Hence information provided to a driver will reflect the conditions that are expected to prevail at network locations at the times when he will actually be there.

A tricky problem in generating anticipatory route guidance is the fact that the system under consideration is affected by the dissemination of information. Indeed, contrarily to weather forecast, the reactions of the users receiving the guidance can affect the future conditions of the network and therefore invalidate the predictions on which the guidance was based. The anticipatory guidance is said to be *consistent* if the predictions on which the guidance is based are the same as those that are forecast to result after drivers react to the guidance.

The consistent route guidance generation problem has been analyzed by several authors from a theoretical viewpoint. In his PhD dissertation, Kaysi (1992) has introduced for the first time the concept of consistency as a solution to the overreaction problem of anticipatory information systems. Other theoretical analysis has been proposed namely by Kaufman et al. (1998) and Engelson (1997). The problem has also been considered in the context of the development of guidance model systems like DynaMIT (Ben-Akiva et al., 1997, Ben-Akiva et al., 2002, Ben-Akiva et al., 1998, Bottom et al., 1999), DYNASMART (Mahmassani et al., 1993) or TRANSIMS (Nagel, 1997).

The first formal formulation has been proposed by Bottom et al. (1999) and developed by Bottom (2000) in his recent PhD dissertation. We summarize here the conceptual framework on which the formulation is based.

This framework is based on exogenous parameters:

- a directed graph representing the traffic network;
- a fixed time horizon, called the *guidance horizon*, divided into a finite number of time intervals, called *update intervals*;
- a description of the transportation demand by origin, destination, departure time, access to information and behavioral class;

- an enumeration of feasible paths in the network;
- a list of decision points where en-route decisions can be implemented;

In this framework the CARG problem can be formulated as a fixed point problem which may be stated as follows:

$$\text{Find } x \text{ such that } x = T(x) \tag{6.1}$$

where x belongs to one of the three sets of endogenous variables:

- *Path flows* P, representing the number of trips of a particular user class traveling from a given decision point to a given destination at a given time.
- *Network conditions* C, that are typically represented by time-dependent link impedances for each time interval in the given time horizon. Practically speaking, in most cases, the impedance is the link traversal time.
- *Guidance messages* M are quadruples involving message type, location, time and contents. The exact definition of these variables depends on the specific technology used. Nevertheless most of the time these variables are elements of a finite set, for example if the messages are particular path recommendations.

Each pair of sets of variables is related by causal relationship, expressed by the following maps and diagrammatically represented in Figure 6.1:

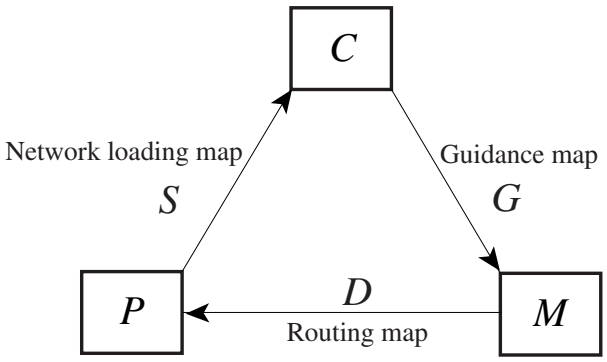


Figure 6.1: CARG: fixed point formulation

- The *Network loading map*, denoted by $S : P \rightarrow C$, determines the traffic conditions that result from the assignment of a given set of time-dependent path flows P over the network.
- The *Guidance map*, denoted by $G : C \rightarrow M$, represents the generation of messages M, based on a given set of traffic conditions C. The map

captures the technological characteristics of the information system. A particular instance of this map, is the identity function which provides directly the raw traffic conditions as information. In this case the CARG problem becomes equivalent to the dynamic traffic assignment problem (cf Bottom et al., 1999).

- The *Routing map*, $D : M \to P$ relates a given set of guidance messages to the resulting path flows. It captures the drivers response to the information, and is typically based on behavioral models such as route choice, departure time choice, and mode choice (see Ben-Akiva and Bierlaire, 1999).

It is important to mention here that both the network loading map and the routing map are usually stochastic. Mainly, the routing map's stochasticity is related to the uncertainty associated to the underlying probabilistic choice model. The network loading map's stochasticity is due to the discrete nature of traffic. We refer to Bottom (2000) for a more detailed analysis of the sources of stochasticity.

The formulation of the consistent anticipatory guidance generation is based on the composition of these three maps. From the fixed point formulation (6.1) T can be described as one these three following compositions:

$$\text{If } x \in P \text{ then } T \text{ is defined by } D \circ G \circ S \tag{6.2}$$

$$\text{If } x \in C \text{ then } T \text{ is defined by } S \circ D \circ G \tag{6.3}$$

$$\text{If } x \in M \text{ then } T \text{ is defined by } G \circ S \circ D \tag{6.4}$$

Remark that these three composite maps are equivalent with respect to the existence of a fixed point: if one has a fixed point then they all do, and if one does not then none does. Unfortunately there is no guarantee that such fixed point exist. Thus in general we will define the *inconsistency* norm of a variable x, belonging to one of the three sets of variables presented above, as the value $\|x - T(x)\|$, for some norm $\|\cdot\|$. We will consider that consistency has been achieved, and roughly speaking the fixed point found, if we find an approximate solution to the CARG problem defined by:

$$\|x - T(x)\| < \varepsilon \tag{6.5}$$

for some small $\varepsilon > 0$. The value of ε depends on the application and the set of variables. For example, an error of 1 minute is fairly acceptable on a 30 minutes trip. Moreover, in practice descriptive informations are most of the time rounded, making the use of very fine criteria meaningless.

The non-existence of a fixed point has two concrete implications on the guidance generation problem. First it means that whatever information is sent to travelers, it will never be consistent with the traffic conditions they will experience, i.e. the guidance is known to have a poor quality before it is sent to

users. Additionally these types of guidance could worsen rather than improve traffic conditions, as shown in (Arnott et al., 1991) and (Hall, 1986). Moreover, from an algorithmic point of view the non-existence of a fixed point invalidates theoretical convergence properties of algorithms used to find it.

6.2.1 Characteristics of the problem

Using this analysis framework, the consistent anticipatory route guidance problem can be described as a fixed point problem with the following characteristics.

- *Large-scale problem*

  The analysis framework proposes three different types of problem variables: path flows, link impedances and information messages. For non-trivial applications, each of these formulations involves a large number of variables. Indeed, the number of variables if $x \in C$ is given by the number of links in the network multiplied by the number of update intervals in the guidance horizon. The number of variables if $x \in P$ is given by the number of paths in the network multiplied by the number of update intervals in the guidance horizon, which for most practical applications is significantly larger than the number of variables if $x \in C$. The variables $x \in M$ may be discrete or continuous, depending on the underlying technology. When x is discrete the CARG is turned into a combinatorial problem. The size of the variables $x \in M$ depends also on the underlying technology.

- *Non closed-form problem*

  The complexity of ITS and the necessity of capturing traveler response to information impose the use of disaggregated models and simulation-based tools. Therefore, the formulation of the fixed point problem cannot be stated as an analytical function. Instead, it can be described only by actually running a set of simulation tools with specific parameters values. As a consequence, techniques that will be invoked may not rely on the availability of derivatives.

- *Stochastic problem*

  The simulation-based approach induces presence of noise in the fixed point computation. The level of stochasticity depends on the simulation tools used to compute the composite map. Bottom (2000) has shown that the level of stochasticity is strongly related to the level of granularity of the traffic's representation. The finer the granularity is, the lower the stochasticity is. Note that real-time systems, like DynaMIT, tend to aggregate vehicles into packets in order to increase the computation speed. This may significantly increase the level of stochasticity of the composite map. Nevertheless the outputs of the composite maps can be

treated not as a realization of stochastic process but rather as a time trajectory of deterministic values affected by noise as stated by Bottom (2000).

- *Real-time requirements*

  Due to the large-scale character of the problem and the sophisticated simulation tools involved in the composite map computation, each evaluation of the objective function is expensive in terms of computation time. For an operational system with real-time response capabilities, the consistent guidance has to be computed in a given time interval. From an algorithmic point of view, it means that the number of function evaluations is limited. Therefore we introduce the concept of *computation budget*. It represents the number of evaluations of the composite map we can afford in the context of real-time applications.

6.3 Existing methods

In this section, we present three different formulations of the problem described in Section 6.2, namely fixed point problems, optimization and resolution of systems of nonlinear equations, with their classical associated resolution methods. Analysis of advantages and drawbacks of each type of methods in the context of the CARG problem are also provided.

6.3.1 Fixed point methods

The formalization of the framework as a fixed point problem enables Bottom (2000) to identify some resolution algorithms designed to solve fixed point problems.

Let $\mathcal{T}$ be a continuously differentiable mapping defined from $\mathbb{R}^n$ to $\mathbb{R}^n$. We seek to solve

$$\mathcal{T}(x) = x. \tag{6.6}$$

A solution x^* of (6.6) is called a fixed point of $\mathcal{T}$.

Definition 1 *Let* $D \subset \mathbb{R}^n$ *and let* $\mathcal{G} : D \to \mathbb{R}^n$. G *is Lipschitz continuous on* D *with Lipschitz constant* γ *if*

$$\|\mathcal{G}(x) - \mathcal{G}(y)\| \leq \gamma \|x - y\| \tag{6.7}$$

for all $x, y \in D$.

Definition 2 *Let* $D \subset \mathbb{R}^n$. $\mathcal{G} : D \to \mathbb{R}^n$ *is a contraction mapping on* D *if* $\mathcal{G}$ *is Lipschitz continuous on* D *with Lipschitz constant* $\gamma < 1$.

Classical algorithms for solving fixed point problems are iterative methods based on the Banach Contraction Principle, which can be expressed as follows:

Theorem 10 *Let* D *be a closed subset of* $\mathbb{R}^n$ *and let* $\mathcal{G}$ *be a contraction mapping on* D *with Lipschitz constant* γ *such that* $\mathcal{G}(x) \in D$ *for all* $x \in D$. *Then there is a unique fixed point of* $\mathcal{G}$, $x^* \in D$, *and the iterative method defined by*

$$x_{k+1} = \mathcal{G}(x_k) \tag{6.8}$$

converges q-linearly to x^* *with q-factor* γ *for all initial iterates* $x_0 \in D$

Remark

1. This theorem is also known as the Contraction Mapping Theorem (Banach, 1922) and has influenced the majority of methods dealing with fixed point problems.

In consequence classical methods to solve fixed point problems can be summarized as follows. Let $\mathcal{T} : X \to X$ a mapping accepting $x^* \in X$ as a fixed point, i.e. $\mathcal{T}(x^*) = x^*$. Choose a starting point $x_0 \in X$ and generate a succession of points of the form:

$$x_{k+1} = x_k + \alpha_k(\mathcal{T}(x_k) - x_k) \tag{6.9}$$

where $\alpha_k \in [0; 1]$. We will refer to *recursive averaging methods* for this class of algorithms (Dunn, 1973, Ishikawa, 1976, Magnanti and Perakis, 1994).

Remark

1. It is interesting to remark that recursive averaging methods defined by (6.9) can be considered as quasi-Newton methods (cf Section 4.2) where B_k^{-1} is defined at each iteration by $B_k^{-1} = \alpha_k I_{n\times n}$, where $I_{n\times n}$ is the identity matrix of size $n \times n$. Indeed, we can write

$$\begin{array}{rcccl} x_{k+1} & = & x_k & - & B_k^{-1} \cdot F(x_k) \\ & = & x_k & + & \alpha_k I_{n\times n} \cdot F(x_k) \\ & = & x_k & + & \alpha_k \cdot (\mathcal{T}(x_k) - x_k)\,. \end{array} \tag{6.10}$$

   It emphasizes the relationship between these two types of methods, illustrating that recursive averaging methods can be considered as a subclass of quasi-Newton methods, where the matrix B_k^{-1} is characterized by $\alpha_k I_{n\times n}$.

   We have performed numerical tests using recursive averaging methods to solve the systems of nonlinear equations presented in Section 5.5. Most of the time these types of algorithms, designed to solve fixed point problems, cannot find the solution of the considered systems.

We now describe some particular instances of this class of method based on the recursive scheme (6.9).

- The *functional iteration method*, also known as the nonlinear Richardson iteration, Picard iteration or the method of successive approximation, is based on the Contraction Mapping Theorem. It starts at an arbitrary point x_0 and applies the iterative step (6.9) with $\alpha_k = 1$ for all k, that is

$$x_{k+1} = \mathcal{T}(x_k) \tag{6.11}$$

  until a fixed point has been approximated to a sufficient degree of accuracy. As stated by Theorem 10 this method converges to a fixed point if $\mathcal{T}$ is a contraction. Ortega and Rheinboldt (1970) have established other sufficient conditions for convergence of this method.

- The simple averaging methods involve the recursion (6.9) where α^k is a sequence such that $\sum_{k=0}^{\infty} \alpha_k = \infty$ and $\sum_{k=0}^{\infty} \alpha_k^2 < \infty$. This class of algorithms was proved to converge to a root of $\mathcal{T}(x) - x$ by Robbins and Monro (1951) and by Blum (1954), despite noisy function evaluations. Chung (1954) and Fabian (1973) also showed that Robbins and Monro recursive averaging procedure is an asymptotically optimal procedure for determining equation roots in the presence of noise. A typical example of a simple averaging method is the *Method of Successive Averages* (MSA) proposed by Sheffi and Powell (1982) where $\alpha^k = \frac{1}{k}$. This method has been successively used for some classical transportation problems, as for example Sheffi and Powell (1982) who used it for stochastic user equilibrium or Cantarella (1997) who applied it to solve two general fixed point formulation of multimode multi-user assignment problem.

  In DynaMIT Ben-Akiva et al. (1997) and Bottom et al. (1999) have developed an algorithm inspired by these simple averaging methods. The method, called Time Smoothing algorithm, uses the recursive scheme (6.9) with a predetermined $\alpha_k \in]0, 1]$ for all k.

- Magnanti and Perakis (1997) and Magnanti and Perakis (1994) propose a scheme to determine α^k by minimizing a potential function $P(\alpha^k)$, using either exact or inexact line search methods. They identify and investigate a variety of potential functions for the fixed point problem in the deterministic case and show that in some cases the theoretical rate of convergence can be faster than averaging methods with predetermined α^k.

- Averaging of iterates proposed by Polyak and Juditsky (1992) refers to computing a running average

$$\xi^k = \sum_{i=1}^{k} \frac{x_i}{k} \tag{6.12}$$

  where x_i are calculated with a simple averaging process (6.9). If the simple averaging process involves a noisy function computation, converges

and if $\alpha_k \to 0$ slower than $o(1/k)$, then Polyak and Juditsky (1992) and Kushner and Yang (1993) proved that the sequence ξ^k converges at an optimum rate. It means that theoretically this procedure equals or surpasses asymptotic performances of any iterative methods defined by any recursive averaging method.

As pointed out by Bottom (2000) the computation of ξ^k can be done off-line using a simple averaging scheme to compute the x_k. But, in practice, the averaging, i.e. the evaluation of the ξ^k (6.12), is only started after the simple averaging iterates show signs of stabilizing, sometimes called the window of averaging. Unfortunately for real-time problems this window of averaging may already exceed the computation budget of the problem.

In his conclusions, Bottom (2000) selects MSA and Polyak's methods as best candidates to solve the consistent route guidance generation problem. When the stochasticity of the composite mapping is low, Polyak's method outperforms MSA in terms of run time efficiency. In the presence of high stochasticity, the performance of both approaches are comparable.

Remarks

1. This kind of methods are useful for large-scale problems due to the insignificant amount of linear algebra associated to the generation of each iterate as well as its moderate memory requirement as only the last two iterates have to be stored.

2. In the case of consistent route guidance generation the mapping $\mathcal{T}$ is given by T, defined by one of the three composite maps ((6.2), (6.3) and (6.4)). As emphasized before, there is no evidence of the existence of a fixed point of T, which invalidates theoretical properties of convergence for these algorithms.

6.3.2 Derivative free optimization methods

Bierlaire and Crittin (2001) have considered the fixed point problem (6.1) as a nonlinear minimization problem:

$$\min_x f(x) \text{ with } f(x) = d(x, T(x)) \tag{6.13}$$

where d defines a suitable distance. Notice that if x^* is solution of problem (6.1) then x^* is a global minimum of f with $f(x^*) = 0$. On the contrary if $\bar{x}$ is solution of (6.13) nothing ensure that $T(\bar{x}) = \bar{x}$. (cf Dennis and Schnabel, 1996). As described above the evaluation of the function is computationally expensive, and so global optimization methods are not adapted for this problem. Moreover as we can not compute the derivatives of the objective function we have to focus on direct search methods. See Bierlaire and Crittin (2001) for a description and criticisms of existing derivative free algorithms as well as a new

method, especially designed for large-scale problems, proposed by the authors. The numerical results obtained on the CARG problem were encouraging as the method decreases significantly the objective function f in the first iterates compared to classical fixed point methods. Unluckily the algorithm often gets stuck in a subspace where no improvement can be achieved anymore, due certainly to the appearance of local minima of the objective function, which are not fixed points of the initial problem. It can explain some mitigated numerical results obtained by these methods.

6.3.3 Methods solving systems of nonlinear equations

Another natural way to express fixed point problems is as resolution of systems of nonlinear equations written:

$$F(x) = 0 \tag{6.14}$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is a continuously differentiable mapping. The equivalence of these two formulations is straightforward. Indeed,

$$x^* \text{ is solution of (6.6)} \Leftrightarrow x^* \text{ solution of (6.14)}$$

simply setting $F(x) = \mathcal{T}(x) - x$.

As described, in Chapter 4 and 5 most methods used to solve problem (6.14), referred as quasi-Newton methods, are iterative methods and can be described as methods using the following iterations:

$$x_{x+1} = x_k - B_k^{-1} F(x_k) \tag{6.15}$$

and update B_{k+1}. The most successful class of quasi-Newton methods are secant methods, but usually they are not designed to solve large-scale systems of equations. The main drawbacks of these methods when they deal with large-scale problems are the storage cost of the matrix B_{k+1} and the difficulties encountered in the resolution of the associated linear system (6.15). As described in Chapter 5 according to Spedicato and Huang (1997), Ruggiero et al. (1996), Luksan and Vlcek (1998) and Martinez (2000) the most effective quasi-Newton method to solve large-scale systems of nonlinear equations is the Inverse Column-Updating Method (ICUM) introduced by Martinez and Zambaldi (1992). Unfortunately, as illustrated by Section 5.5.2, starting quasi-Newton process with an arbitrary matrix very often causes catastrophic results. The computation of an initial good approximation of the Jacobian is almost always necessary. As a consequence, in the context of the CARG problem, these algorithms are useless because of the impossibility to compute such a good approximation without a big computing effort. Moreover large-scale implementations of ICUM needs a structured Jacobian in order to efficiently factorize the matrix B_{k+1}^{-1} and so avoid its explicit construction, which can be deterrent for large-scale applications. In practice, the most efficient approach to solve

large-scale problems are inexact-Newton algorithms. Among these methods the Newton-Krylov class of algorithms described in Section 5.2.2, uses, as its name suggests, Krylov subspaces based on linear solvers to solve (5.8). This family of algorithms, allowing to perform only matrix-vector products is especially well adapted for large-scale problems as underlined by Kelley (2002b). Newton-Krylov methods use partial derivatives information estimated using finite differences. Consequently they are of no help in the context of the CARG problem. Indeed, the noise present in the evaluation of the composite mapping prevents the method from using finite differences, as demonstrated by the behavior of these methods in presence of noise in Section 5.6.

This review of the classical methods to solve systems of nonlinear equations underlines the need for a more specific method to solve the consistent anticipatory route guidance, in particular due to the presence of noise in the evaluation of the objective function and the necessity of a matrix-free algorithm to deal with the high dimension of the variables. The algorithm presented in Chapter 5 has been designed to handle the intrinsic characteristics of the CARG problem.

First it is a matrix-free algorithm allowing to solve problems with high dimension. Second this method does not use finite differences to compute approximation of derivatives information, but uses only the previous iterates as the basis to compute the next iterate. This feature is particularly attractive in the context of noisy problems and when the function is expensive to evaluate as we use all the available information collected through the last iterates.

Following the notations of Chapter 5 iGSM is an iterative method that can be summarized as follows, at each iteration:

$$x_{k+1} = x_k + p_k \tag{6.16}$$

where p_{k+1} is computed using the following matrix-free procedure:

$$\text{Solve } z = \underset{y}{\operatorname{argmin}} \|Ay - F(x_{k+1})\|_2^2 \tag{6.17}$$

$$\text{Define } p_{k+1} = -Cz \tag{6.18}$$

where $A = (Y_{k+1}\Omega \quad \Gamma)$ and $C = (S_{k+1}\Omega \quad \Gamma\widehat{H}^0_{k+1})$. The least squares (6.17) is solved using a matrix-free conjugate gradient method called LSQR (Paige and Saunders, 1982) especially designed to solve large-scale and sparse problems.

In Section 5.4 the choice of the matrix $\widehat{H}^0_{k+1} = I_{n\times n}$ has been motivated by numerical reasons looking for a trade off between efficiency and robustness of the method. In the context of the search of a fixed point this choice is even more obvious. The role of $\widehat{H}^0_{k+1}$ is to overcome the possible underdetermination of the least squares (5.11). It can be considered as an a priori approximation of B^{-1}_{k+1}. Meaning that, by default and without any information about the objective function, iGSM is close to recursive averaging methods as described by equations (6.10). Indeed, it computes B^{-1}_{k+1} trying to be as close as possible to $\alpha_{k+1} I_{n\times n}$ in a least-squares sense.

Due to the impossibility of computing a good approximation of the Jacobian of $F(x_0) = T(x_0) - x_0$, and following the same argument as above, we set initially $H_0^{-1} = I_{n \times n}$ (cf Section 5.4). In this way the first iterate of iGSM is equivalent to the functional iteration described by (6.11). Indeed,

$$x_1 = x_0 + H_0^{-1} F(x_0) \tag{6.19}$$

$$= T(x_0). \tag{6.20}$$

6.4 Numerical results

We analyze the performance of iGSM algorithm compared to recursive averaging methods used to solve the CARG problem. Two different software systems are used to generate anticipatory route guidance. In Section 6.4.1 we use a simulation tool implemented by Bottom (2000) to illustrate the behavior of these algorithms on a small synthetic network. Then, in section 6.4.2, different case studies are presented to compare these algorithms on real networks using the DynaMIT traffic simulation software described in Chapter 2.

Proposed by Bottom (2000), the measure of inconsistency (6.5) seems to be a natural stopping rule for these iterative methods. In practice the identification of ε is a well-known difficulty. Moreover in the context of real-time applications, the number of iterations necessary to reach a given level of inconsistency will most of the time exceed the available computation budget. Consequently in the following we simply stop the algorithms when the number of function evaluations reaches a predetermined computation budget. The objective is to find the most consistent guidance within that budget.

All guidance generation calculations involve, for the both systems softwares, the composite map (6.3), defined over the network conditions set of variables. In this case the fixed point problem is to find $x \in C$ such that

$$x = S \circ D \circ G(x). \tag{6.21}$$

The variables of this problem are the time-dependent link impedances for all update intervals in the given guidance horizon. Formulation (6.21) involves the smallest number of variables compared to the composite map (6.2) defined over the path flows. Indeed, the number of links of a network is generally smaller than the number of paths defined between each origin-destination pair. For example the Swiss network described in Section 6.4.2 counts 1661 links, but 91'226 paths. For the other composite map (6.4), based on the guidance messages, the fixed point remains well defined, but its determination will represent a large combinatorial problem.

iGSM has been implemented in *C++* based on an object-oriented analysis and introduced in both system softwares. For the linear algebra a library named *Template Numerical Toolkit (TNT)* (Pozzo, 1998) has been used. The implementation of the algorithm follows the description given in Section 5.4. The size of population is set to 5 due to the large dimension of these problems.

Note

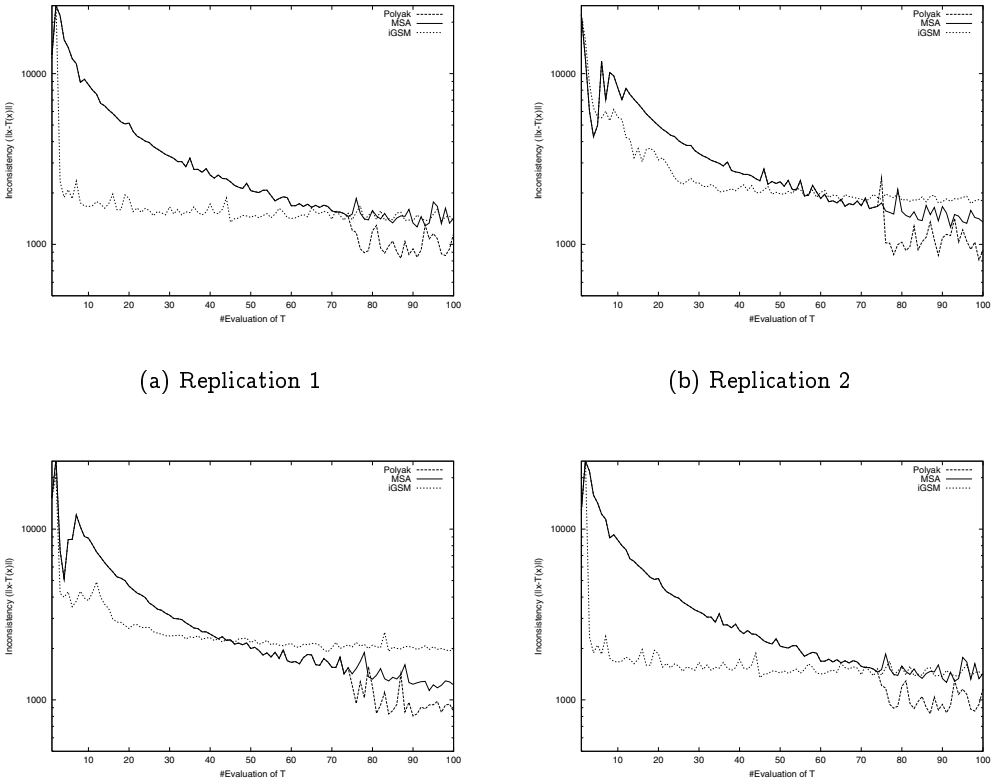
- With this $C++$ implementation of iGSM we have been able to solve a problem proposed by Robert and Shipman (1976) ([SPED27]) with 1 million of variables. It clearly demonstrates the applicability of iGSM to very large-scale problems.

6.4.1 Simple Simulator

We present here the results obtained for the anticipatory route guidance generation using a simulator implemented by Bottom (2000). This code implements in C++ a simple version of the network loading, routing and guidance components maps described in Section 6.2. This software is intended as a test bed for investigating different problem formulations and resolution methods for the consistent anticipatory route guidance generation. The presented runs were made using a simple 14-links network with a single OD pair and eleven OD paths illustrated by Figurein 6.3. The evolution of the network is simulated on a guidance horizon of 40 minutes with update intervals of 1 second. We also assume that 50 % of drivers have access to information, with a demand rate of 10800 trips per hour from origin to destination over 20 minutes (cf Bottom, 2000 for a more detailed description of the simulator). With these specifications the size of the fixed point problem is 33600 (14×2400). We generate four replications, all starting with the table of free-flow link traversal times as starting point. In practice Polyak iterate averaging method is started after MSA stabilizes. In this case Polyak's method is ran after performing 75 steps of MSA. The computation budget has been set to 100 function evaluations in order to be able to run the Polyak's method.

Figure 6.2 contains four replications, each figure gives the value of the inconsistency of each algorithm against the number of function evaluations for iGSM, MSA and the Polyak's method. For MSA and Polyak's methods these results confirm the numerical results obtained by Bottom (2000).

Clearly iGSM performs well in the first iterates of each replication, a good feature considering real-time applications, but struggles afterwards. MSA and Polyak start slower than iGSM but they are able to decrease more the inconsistency in the last iterates, especially on Figure 6.2(c). It can be explained by the fact that these averaging methods are designed to asymptotically handle the stochasticity, allowing to perform very small steps for each iterate and so accomplishing a kind of implicit filtering of the iterates. This procedure seems to be effective, mainly with low values of the inconsistency. This is especially noticeable from iteration 75. Polyak's method is really efficient and decreases even more the value of the inconsistency, as already pointed out by Bottom (2000). Unfortunately the Polyak's algorithm is not designed for real-time applications with tight budget constraints, i.e. problem where maximum number of function evaluations allowed is small, since this method can only be started



(a) Replication 1

(b) Replication 2

(c) Replication 3

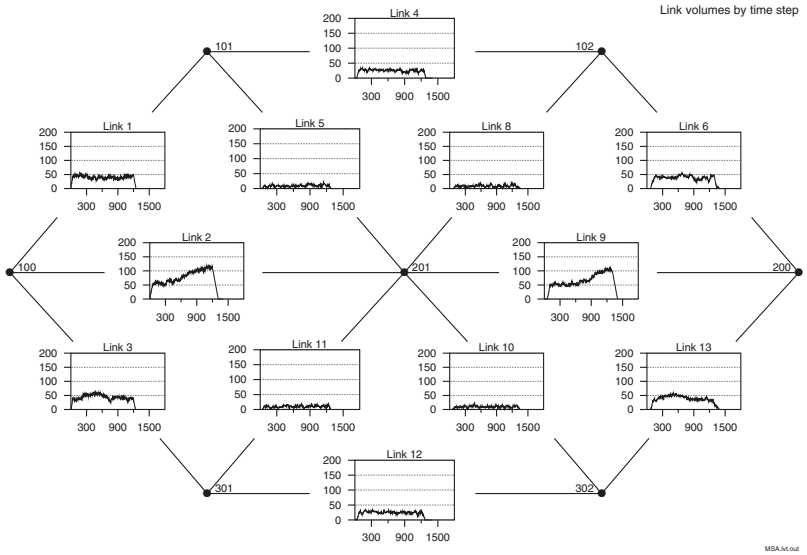
(d) Replication 4

Figure 6.2: Simple Simulator

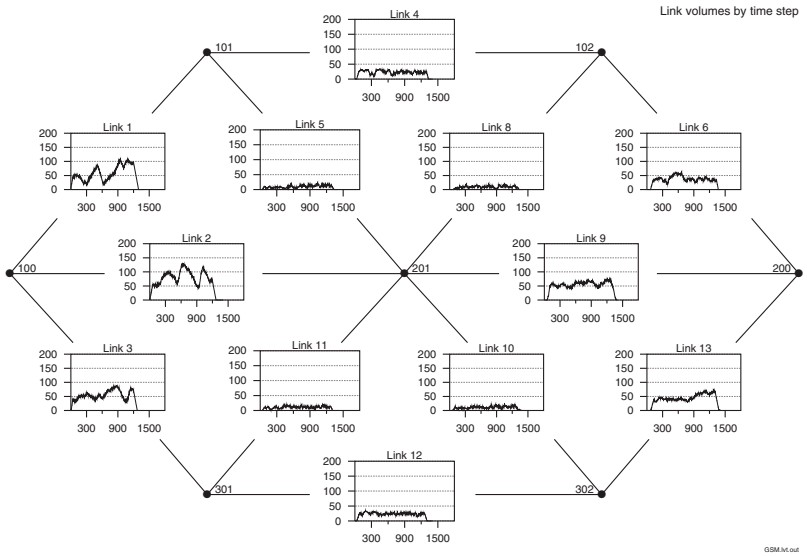
after the stabilization of the iterates computed by the averaging method used to compute the x_k in (6.12).

Figure 6.3 gives an interpretation of the solution obtained by MSA and iGSM for Replication 1, in terms of density of traffic for the whole guidance horizon. The graph plotted on each link represents the link volume, given in vehicles, by update interval. It is interesting to notice that the guidance generated by MSA, which has almost the same consistency as the one generated by iGSM (see Figure 6.2(a)), produces more regular flows on links 1,2 and 3. iGSM seems to spread more erratically the flows on these three links. Moreover comparing Figure 6.3(a) and 6.3(b) on links 2 and 9 shows that MSA sends more vehicles on these twos links than iGMS, which prefers to spread it through the entire network. These observations tends to point out that the same level of inconsistency can be produced by different guidances, influencing the distribution of the road users and consequently the state of the network. This remark underlines the necessity of considering other criteria, in addition to the inconsistency, to generate anticipatory route guidance. At least in order to intelligently choose, from a user's optimal point of view as well as a system

optimal point of view, between two guidances with almost the same level of inconsistency.



(a) MSA



(b) iGSM

Figure 6.3: Link volume trajectories (Replication 1)

6.4.2 DynaMIT

As described in Chapter 2 DynaMIT is a state-of-the-art, real-time computer system for traffic estimation prediction and generation of traveler information. It is designed to operate in real time, using traffic volume and control system state data to estimate and predict time-dependent origin-destination flows and network conditions, and generating descriptive and prescriptive information that should be consistent with the predicted traffic conditions. DynaMIT's consistent guidance generation algorithm is currently the Time Smoothing algorithm described in section 6.3.1. We have implemented MSA and iGSM algorithms in this system in order to compare their efficiency.

DynaMIT uses a rolling horizon framework. The system computes consistent guidance over the guidance horizon (cf Section 6.2), for example, every 15 minutes. This time interval is called the *roll stage*. After consistent guidance is computed for the entire guidance horizon, the system disseminates messages at each update interval. Notice that only the messages for times up to the guidance transmission in the next roll stage need to be disseminated. At this point it moves forward in time by one roll stage and compute the guidance for this new time horizon.

For large networks, the evaluation of the composite map on computer equipped with a 2.4 GHz CPU takes a few minutes. We have arbitrarily chosen a computation budget of 20 evaluations of the composite map as generally the time horizon to generate a guidance is 15 minutes with time intervals of one minute.

Florian network

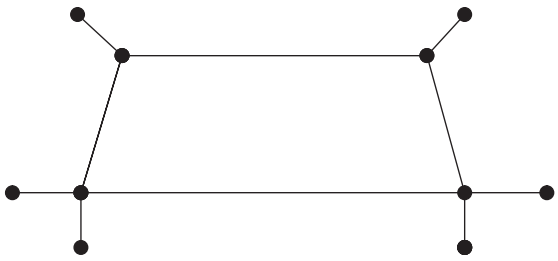


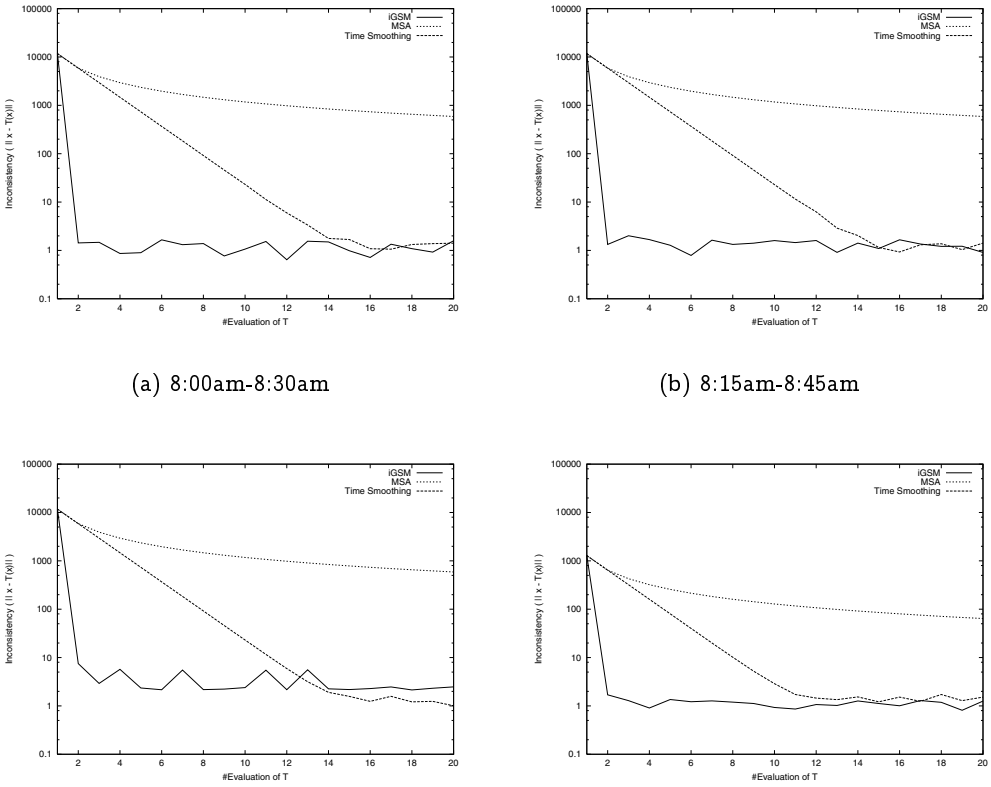
Figure 6.4: Florian network

The first network, called Florian, is a miniature synthetic network, illustrated by Figure 6.4, composed of 10 links.

The guidance is produced for a guidance horizon of 30 minutes with update intervals of 1 minute. The roll stage is set to 15 minutes. The simulation is run from 8:00am to 9:15am. In short consistent guidance is computed for time intervals 8:00am-8:30am, 8:15am-8:45am, 8:30am-9:00am and 8:45am-9:15am.

The size of the fixed point problem associated to the CARG problem for this tiny network is 300 (30×10).

As shown in Figure (6.5) a stationary value for the consistency is nearly reached in 3 iterations using iGSM but it needs more than 10 iterations using the Time Smoothing algorithm to obtain the same consistency, and even more for MSA. We cannot draw too much conclusions from this synthetic small problem, but it is interesting to remark that the behavior of the considered methods is similar to the behaviors obtained in Section 6.4.1.



(a) 8:00am-8:30am

(b) 8:15am-8:45am

(c) 8:30am-9:00am

(d) 8:45am-9:15am

Figure 6.5: CARG on Florian network

Central Artery Tunnel network

The second network is the Central Artery/Third (CA/T) Harbor Tunnel network, currently under construction at Boston. It is a real medium-scale network, with 211 links and 10 OD pairs, represented in Figure 3.5.

The guidance is produced for only one guidance horizon of 30 minutes, from 7:45am to 8:15am, with update intervals of 1 minute. The size of the fixed point problem associated with the CARG for this network is 6330 (211×30).

For this network iGSM reaches the solution at the second iterate, more precisely $\|x_1 - T(x_1)\| < \varepsilon$ with $\varepsilon = 10^{-5}$. As described in Section 6.3.3 the first iterate computed by iGSM is a functional iteration, meaning that $\|T(x_0) - T(T(x_0))\| < \varepsilon$. In Table 6.1, the series of the inconsistency values obtained by the Time Smoothing algorithm and MSA are also reproduced.

	Inconsistency ($\|x - T(x)\|$)		
#Function evaluations	MSA	Time Smoothing	iGSM
1	1115.14	1115.14	1115.14
2	557.57	557.57	0.00
3	371.71	278.78	
4	278.78	139.39	
5	223.02	69.69	
6	185.85	34.84	
7	159.30	17.42	
8	139.39	8.71	
9	123.90	4.35	
10	111.51	2.17	
11	101.37	1.08	
12	92.92	0.54	
13	85.78	0.27	
14	79.65	0.13	
15	74.34		
16	69.69		
17	65.59		
18	61.95		
19	58.69		
20	55.75		

Table 6.1: CA/T 7:45am-8:00am

Irvine network

This network contains the major highways I-5, I-405 and CA-133 around Irvine, Ca. It contains also arterial roads in a triangular area defined by I-5, I-405 and Jeffrey road (see Figure 3.6). It is composed of 618 links, 296 nodes and 627 OD pairs. Irvine network data comes from a traffic management center in Irvine, California. 618 The simulation is run from 4:00am to 8:00am. The guidance horizon is a 45 minutes interval with update intervals of one minute. The roll stage is set to 15 minutes producing 14 instances of the problem. The size of the fixed point problem associated with the CARG problem on this network is 46'350 (618×75).

Contrarily to the other networks described in this chapter using free flow link traversal time as initial iterate for the CARG problem, the first iterate is different here. The demand data has been calibrated using a static equilibrium approach and the starting point for the CARG problem, i.e. a link traversal time table, has been computed using the solution of this equilibrium problem. We write this starting point x_0^A. The sequences of inconsistency values computed by iGSM, MSA and Time Smoothing algorithm on each of the 14 periods of time are given in Appendix 6.6.1. The behavior of the algorithms

starting using this starting point is different from the behavior obtained on the other networks. It is interesting to notice that with a starting point producing low inconsistency all algorithms have difficulties to significantly decrease the inconsistency, especially when the demand is low early in the morning (see Figure 6.10 to Figure 6.15). For example, for the period of time 4:45am-5:30am given in Figure 6.13, none of the algorithms is able to significantly decrease the initial value of the inconsistency. It could be explained by the fact that the starting point obtained using the solution of the static equilibrium problem is a good approximation of the fixed point associated to the CARG problem, especially when the demand is low. When the demand increases this approximation is less precise and the algorithms are able to slightly decrease the value of inconsistency, as illustrated by Figures 6.19 to 6.23.

To analyze the impact of quality of the starting point on the performance of the algorithms used to solve the CARG problem, we have also considered another starting point, noted x_0^B. It is the link traversal time described above multiplied by ten, i.e. $x_0^B = 10x_0^A$. This new starting point produces a high value for the inconsistency. The results obtained on the 14 periods are plotted in Appendix 6.6.2. Interestingly the shape of the inconsistency sequences generated is now similar to the shape obtained on the other networks starting with the free flow link traversal time. Like for the other networks iGSM decreases the value of the inconsistency fast compared to MSA and the Time Smoothing algorithm. Note that in Figure 6.33 the inconsistency evaluated at the first iterate is not same between iGSM or the Time Smoothing method and MSA. It is due to the impact of the previous guidances on the state of the network due to the use of the rolling horizon. More precisely, for each roll stage the algorithms compute a guidance which is disseminated to the users. Each guidance influences the future state of the network and consequently the computation of the guidance for the next time horizon. As MSA produces guidances with a higher level of inconsistency in comparison with iGSM and the Time Smoothing algorithm, the state of the network can be really different at the start of a given guidance horizon, explaining these differences of inconsistency at the first iteration.

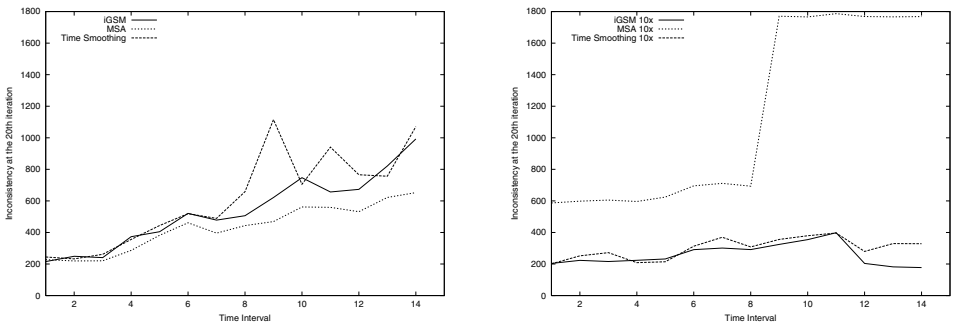
To compare the global results obtained with these two different starting points x_0^A and x_0^B Figure 6.6 plots for the 14 time horizons the value of inconsistency obtained by the algorithms at the 20th iteration.

Figure 6.6(a) gives the results obtained by each algorithms starting with x_0^A. It is noticeable that the inconsistency increases with time for all methods. Nevertheless, globally MSA obtained the best results.

Figure 6.6(b) gives the results obtained by each algorithm starting with x_0^B. In this case the Time Smoothing algorithm and iGSM are really comparable in terms of performance, despite the lower inconsistency obtained by iGSM for the last period of time. The poor results of MSA are due to the influence of the previous guidance producing network conditions where the inconsistency, already at the first iterate, is worse than that computed by iGSM or the Time

Smoothing algorithm.

Comparing Figure 6.6(a) and Figure 6.6(b) bring out an interesting phenomenon. The inconsistency obtained with x_0^A are worst than those obtained with x_0^B. Thus a better starting point for solving the CARG problem may not produce better results, on the contrary. This can perhaps be explained by the fact that the starting point x_0^A coincides with a local minimum of the inconsistency function ($\|x - T(x)\|$), when a much worse starting point, as x_0^B , allows the algorithms to find a another local minimum of the function with a lower inconsistency. At this point a more detailed analysis is needed to verify this hypothesis. This remark about the influence of the starting point is important in practice as the CARG problem will be performed frequently on the same network. The solution of a previous instance of the problem can be considered as a good approximation of the next instance, but will not necessarily produce a more consistent guidance. At this point a more detailed analysis is needed to verify this hypothesis.



(a) Equilibrium starting point

(b) 10x starting point

Figure 6.6: Global results on Irvine network

Swiss network

This interurban network consisted of highways and cantonal routes represents the western Switzerland highway system from Geneva to Schaffhausen given in Figure 6.7. This network counts 11'088 OD pairs. It is composed of 1661 links and 91'226 paths. It motivates the choice of the composite map considering the network conditions as variable of interest for the CARG problem rather than the path flows. The demand is a mixed between international, national (intercity), and local (around main cities). The total historical demand is 932'650 trips/day.

The simulation is run from 7:00am to 8:45am. The guidance horizon is 75 minutes interval with update intervals of one minute. The roll stage is set to 15 minutes producing three instances of the problem. The size of the fixed point problem associated with the CARG problem on this network is 124'575

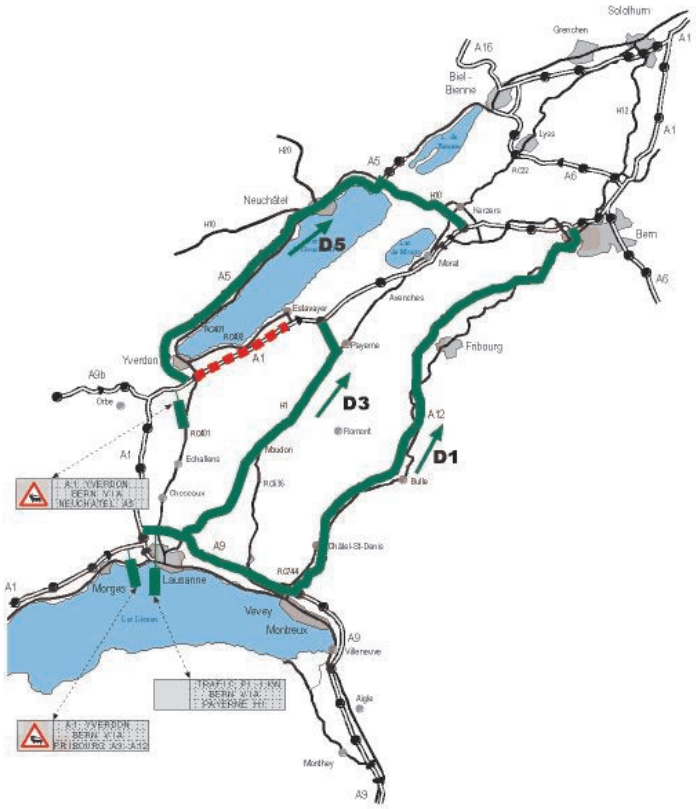


Figure 6.7: Part of the Swiss network

(1661×75). The initial point for all algorithms is the table of free-flow traversal times.

Once more the behavior of algorithms in harmony with the results obtained with the other networks as illustrates by Figure 6.9: a quick decrease of the objective function by iGSM in the first iterates followed by a stagnation. The time smoothing algorithm reaches the same level of inconsistency a few iterations later, but also struggles at this level. MSA shows more difficulty to decrease the inconsistency quite as much with 20 function evaluations, especially for the last interval as shown in Figure 6.9(c).

To illustrate the impact of noise of the composite map Figure 16.8 exhibits 3 replications of the algorithm iGSM on the Swiss network for the period 7:00am-8:15am. It is only when the algorithm remains at the same level of inconsistency

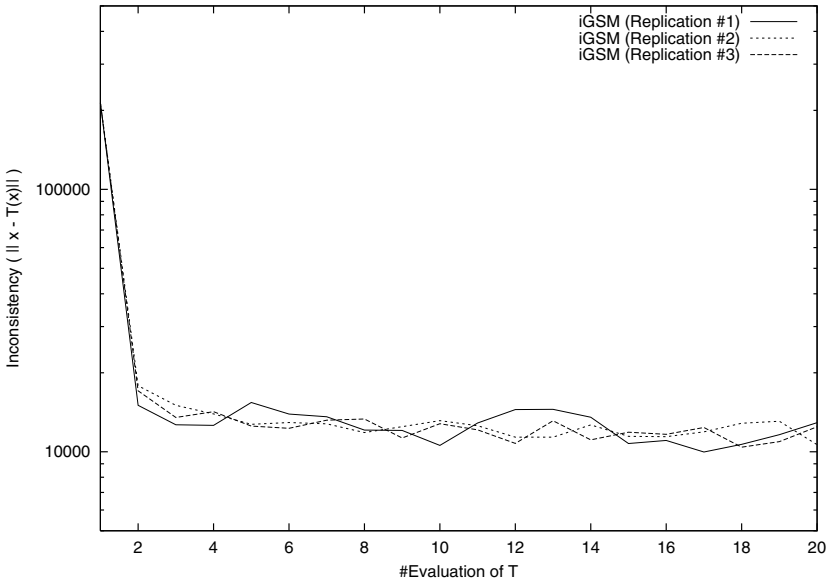
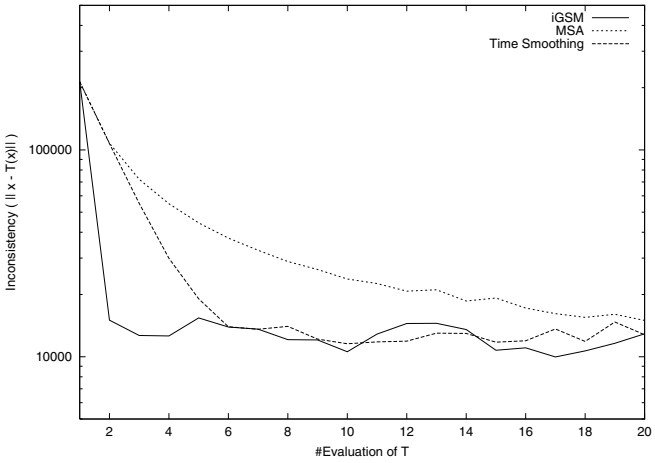


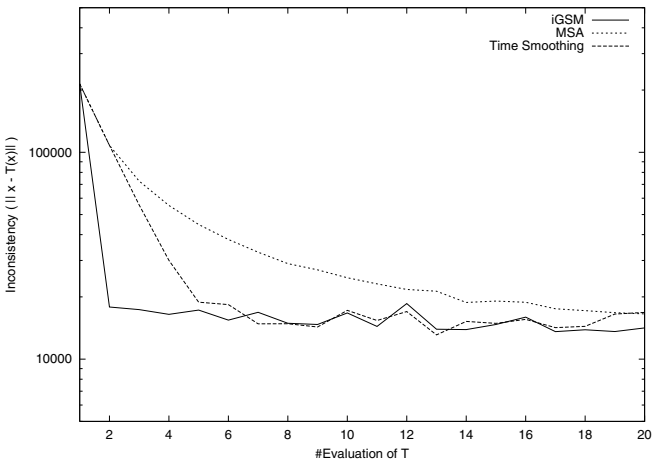
Figure 6.8: Swiss network - 3 Replications

that the noise slightly perturbs the algorithms. We have observed similar behaviors with other algorithms and other instances of the CARG problem computed using DynaMIT.

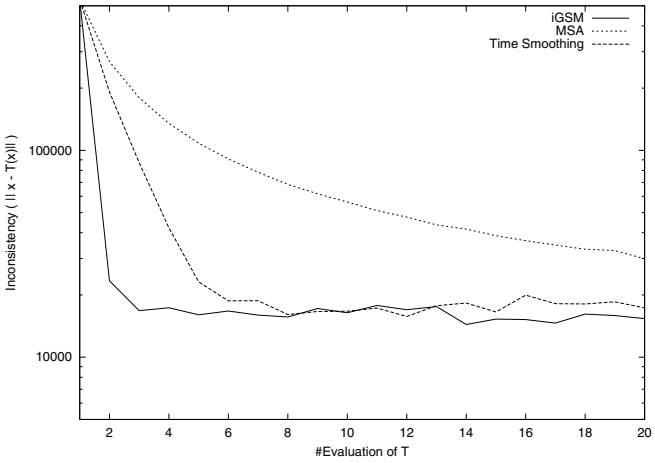
Those preliminary results on the consistent anticipatory route guidance problem are very encouraging, principally for real-time applications, even if a deeper analysis is required to better understand the algorithms behaviors.



(a) 7:00am-8:15am



(b) 7:15am-8:30am



(c) 7:30am-8:45am

Figure 6.9: Swiss network

6.5 Conclusion

In this chapter we have proposed a new formulation of the CARG problem in terms of the resolution of a system of nonlinear equations based on the analysis framework given by Bottom (2000). This formulation allows us to consider new resolution algorithms. Because of the particular characteristics of this problem, mainly the large size and the noise associated to the evaluation of the objective function, standard methods are unsuitable. We have proposed to use iGSM described in Chapter 5 to solve this problem.

Numerical results have been provided using two simulation softwares. A traffic simulation software system, implemented by Bottom (2000), has allowed to bring out some distinctive features of the CARG problem. Especially the difference in terms of the state of the network obtained using two guidances with the same level of inconsistency computed by two different algorithms. This simple simulator also demonstrates the good behavior of iGSM in the first iterates compared to averaging methods presently used to solve the CARG problem.

DynaMIT has been used on a number of case studies. The performances of iGSM are promising, particularly in the context of real-time applications, as it decreases more rapidly the inconsistency than recursive averaging methods.

The numerical results also open new fields of investigations for the generation of consistent anticipatory route guidance itself, in particular:

- Further developments seem necessary to clearly understand the role played by the starting point on the performance of the different resolution algorithms. We have shown in this study that a starting point with a low value of inconsistency does not produce better guidance than a starting point with a high value of inconsistency. This observation is counter-intuitive, as when the traffic conditions do not vary drastically, the solution of previously solved instances of the problem should be a good approximation of the solution of the next instance. Therefore investigations of the influence of the starting point on the generation of consistent guidance would have consequential influences on the real-time applications.

- Additional characterizations of the CARG problem should be investigated. In particular the interpretation of solutions supplied from different algorithms, illustrated by the differences in terms of traffic pattern given by two different guidances with the same level of inconsistency. Moreover we can suppose that sometimes the CARG problem might have multiple approximate fixed point solutions. In that case also we need to be able to select a guidance from among various solutions using other criteria in addition to the inconsistency. The two other composite maps define by (6.3) and (6.4) that characterize the fixed point formulation could certainly produce good indicators.

- In this chapter we have investigated only one of the three proposed composite maps of the problem, namely the composite link condition map. Other formulations need to be investigated, as for example the composite path time map, which appeared to involve the least amount of stochasticity and hopefully allows the algorithms to perform better.

We have strong feelings that solving the consistent anticipatory route guidance problem with a system of nonlinear equations approach rather than a fixed point approach will allow to really speed up the process of generation of consistent guidance particularly in the context of real-time applications.

6.6 Appendix

6.6.1 Irvine (*equilibrium starting point*)

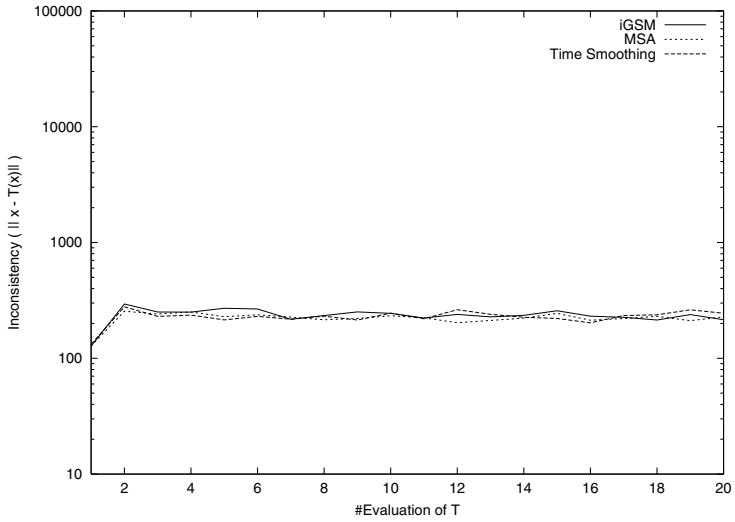


Figure 6.10: Irvine network ($x_0^{\vec{A}}$) 4:00am-4:45am

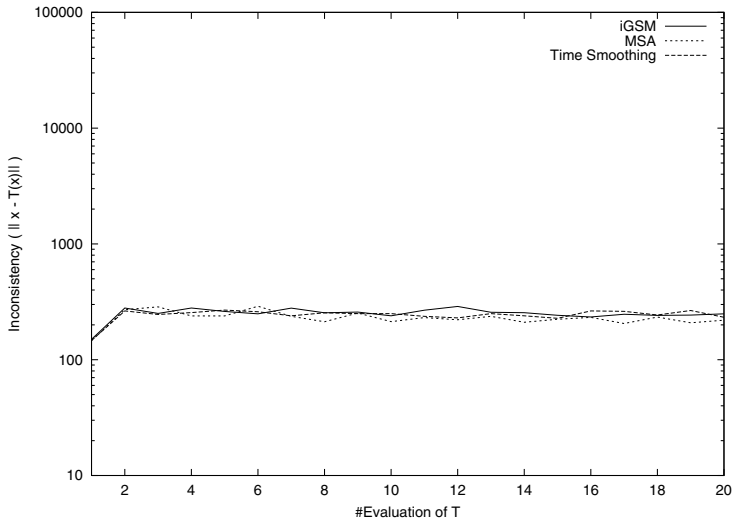


Figure 6.11: Irvine network (x_0^A) 4:15am-5:00am

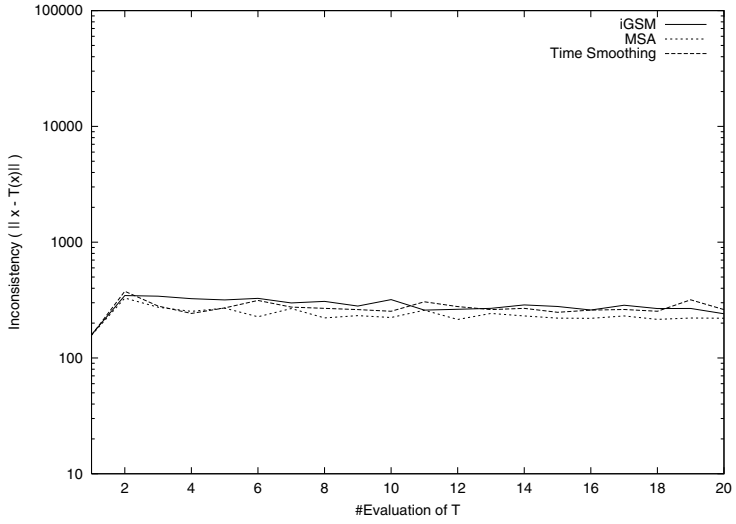


Figure 6.12: Irvine network (x_0^A) 4:30am-5:15am

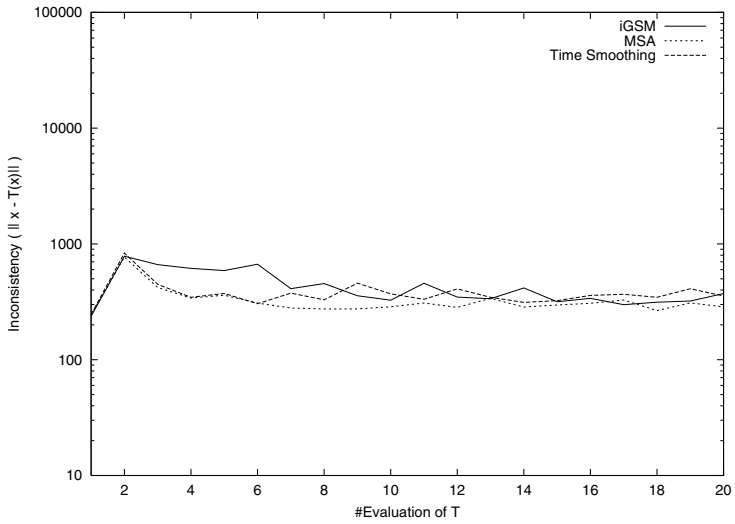


Figure 6.13: Irvine network (x_0^A) 4:45am-5:30am

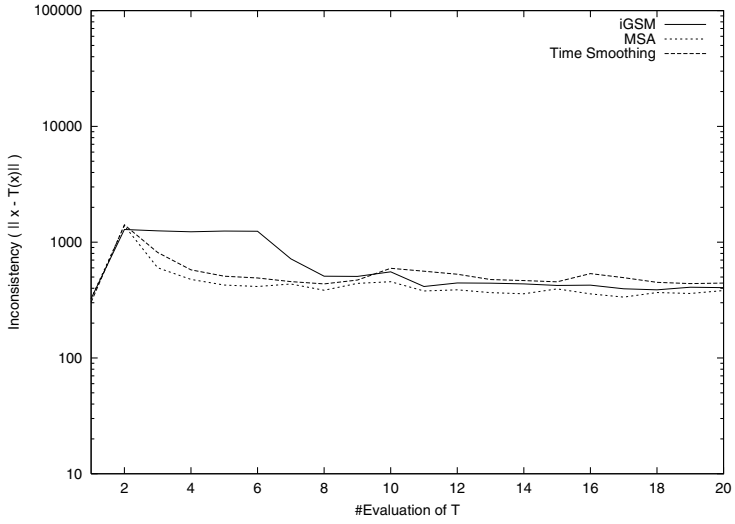


Figure 6.14: Irvine network (x_0^A) 5:00am-5:45am

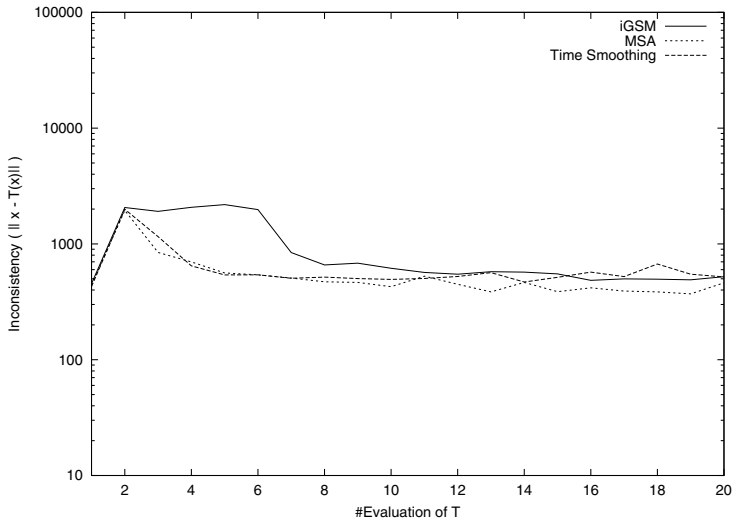


Figure 6.15: Irvine network (x_0^A) 5:15am-6:00am

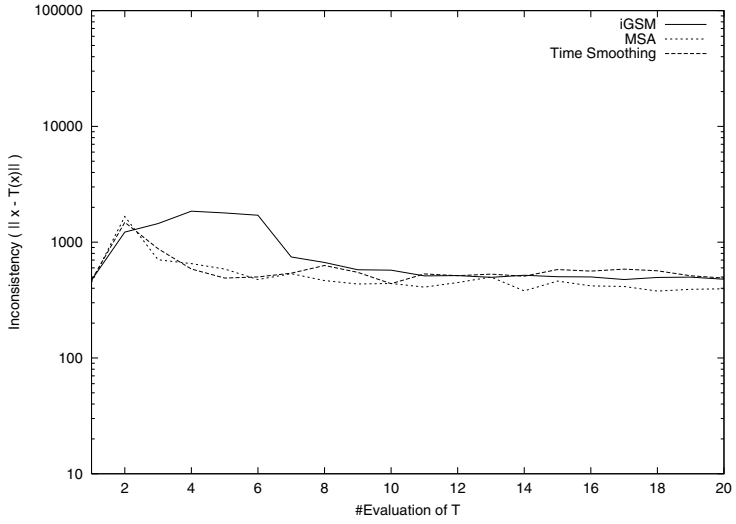


Figure 6.16: Irvine network (x_0^A) 5:30am-6:15am

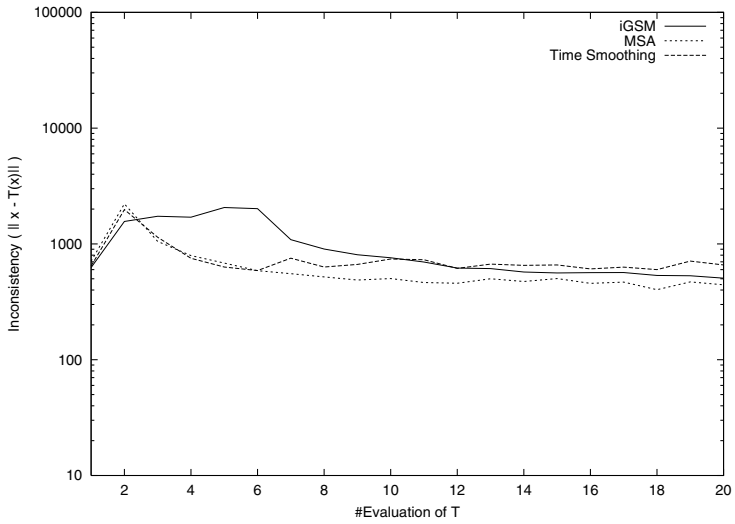


Figure 6.17: Irvine network (x_0^A) 5:45am-6:30am

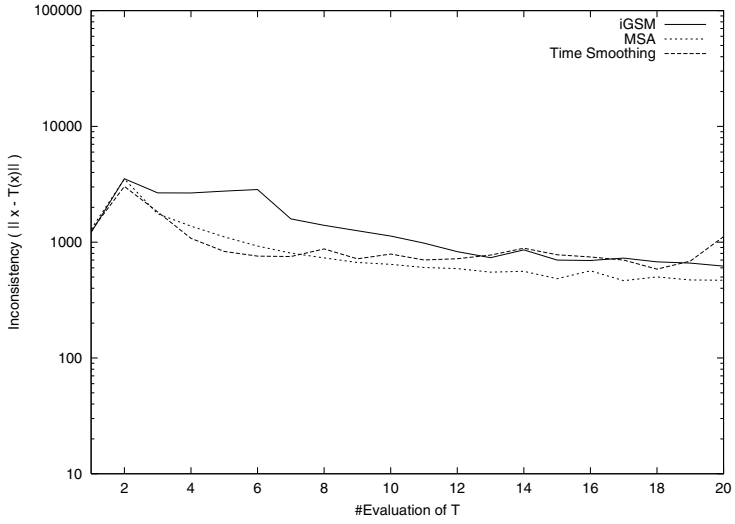


Figure 6.18: Irvine network (x_0^A) 6:00am-6:45am

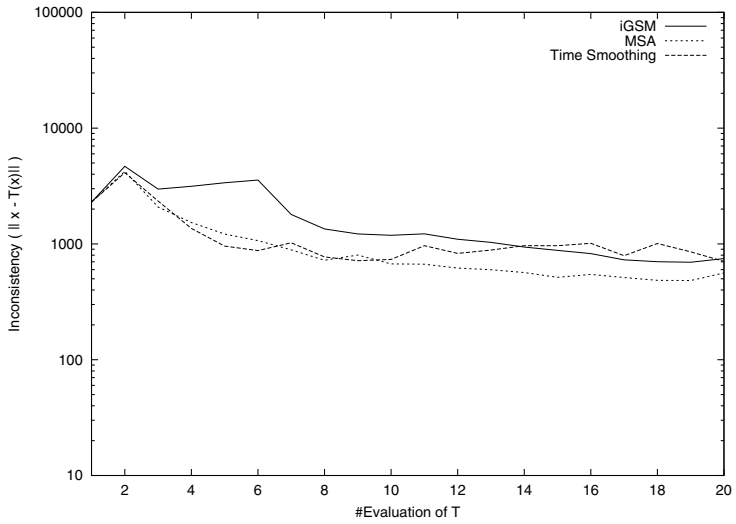


Figure 6.19: Irvine network (x_0^A) 6:15am-7:00am

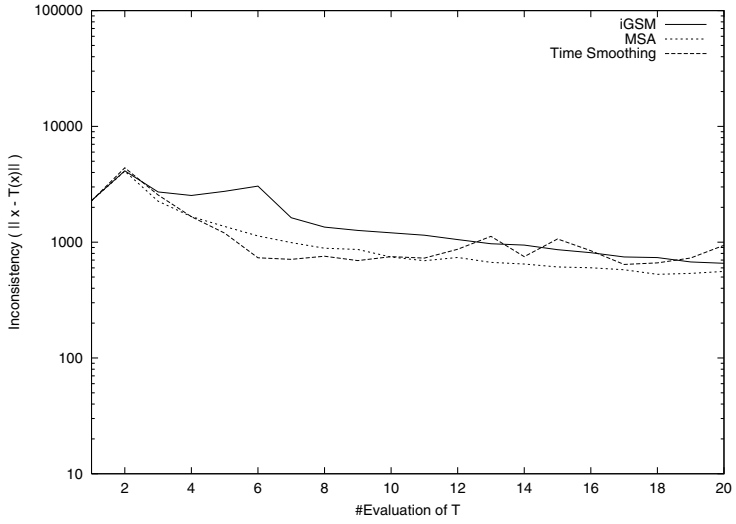


Figure 6.20: Irvine network (x_0^A) 6:30am-7:15am

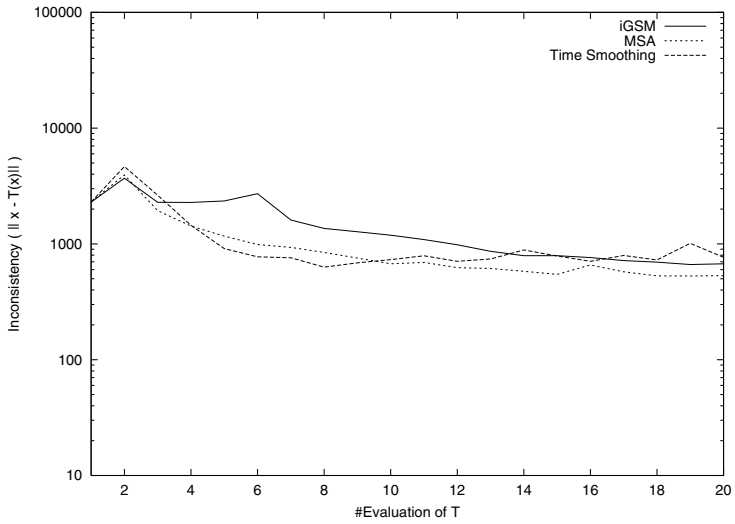


Figure 6.21: Irvine network (x_0^A) 6:45am-7:30am

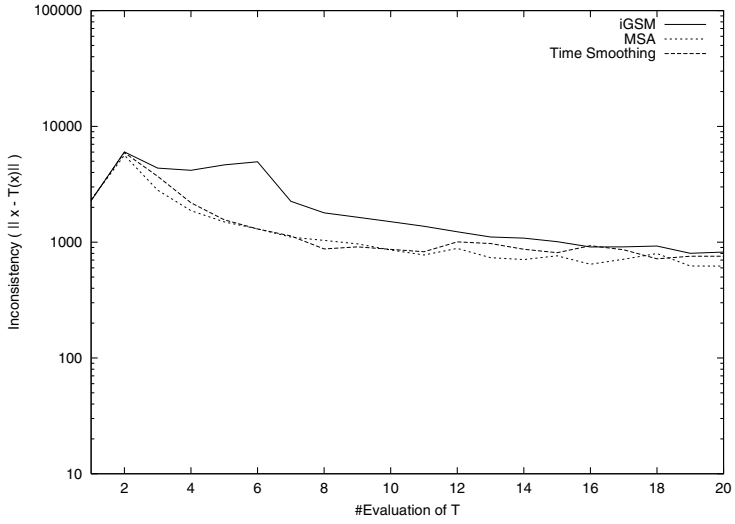


Figure 6.22: Irvine network (x_0^A) 7:00am-7:45am

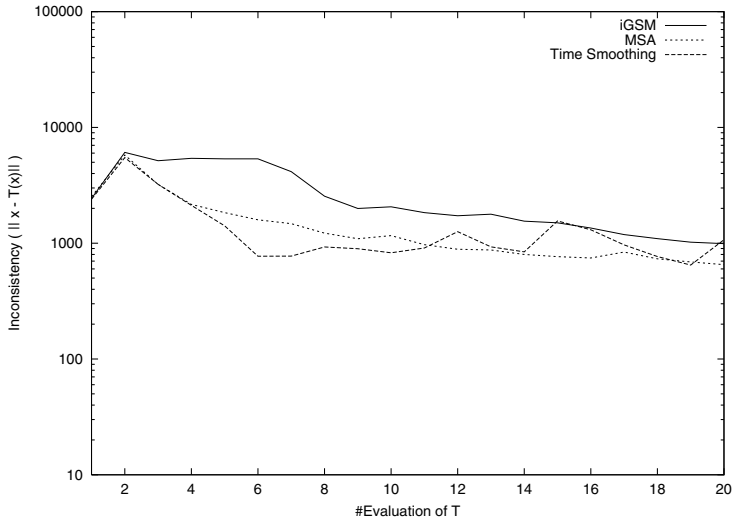


Figure 6.23: Irvine network (x_0^A) 7:15am-8:00am

6.6.2 Irvine (*10 times equilibrium starting point*)

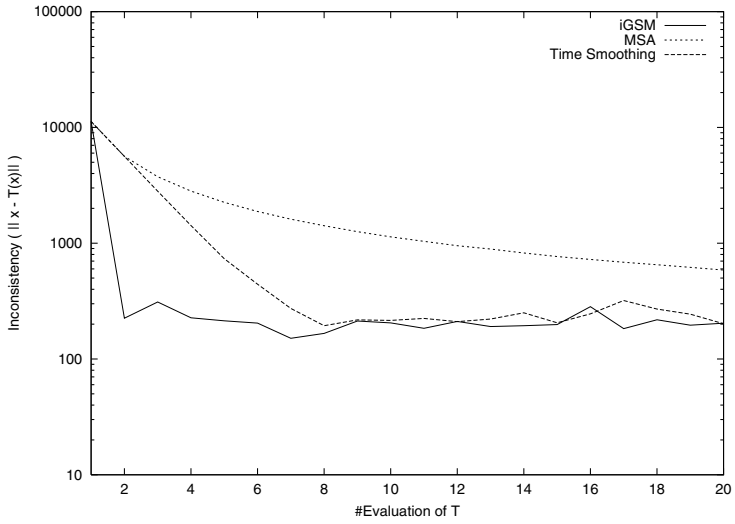


Figure 6.24: Irvine network (x_0^B) 4:00am-4:45am

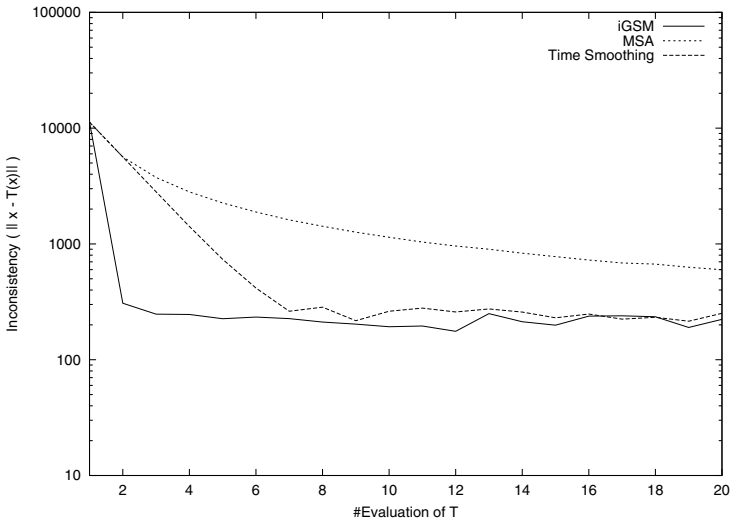


Figure 6.25: Irvine network (x_0^B) 4:15am-5:00am

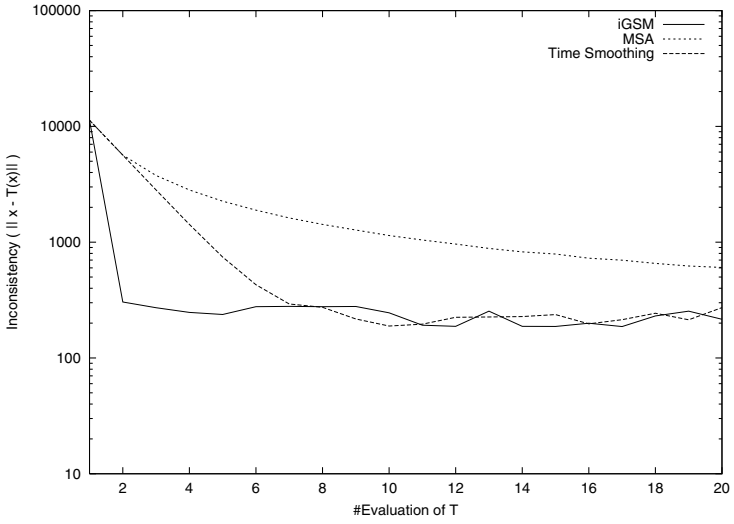


Figure 6.26: Irvine network (x_0^B) 4:30am-5:15am

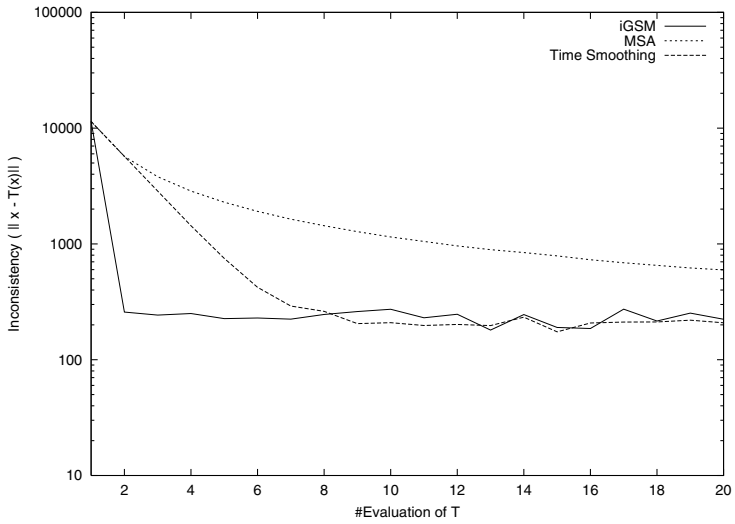


Figure 6.27: Irvine network (x_0^B) 4:45am-5:30am

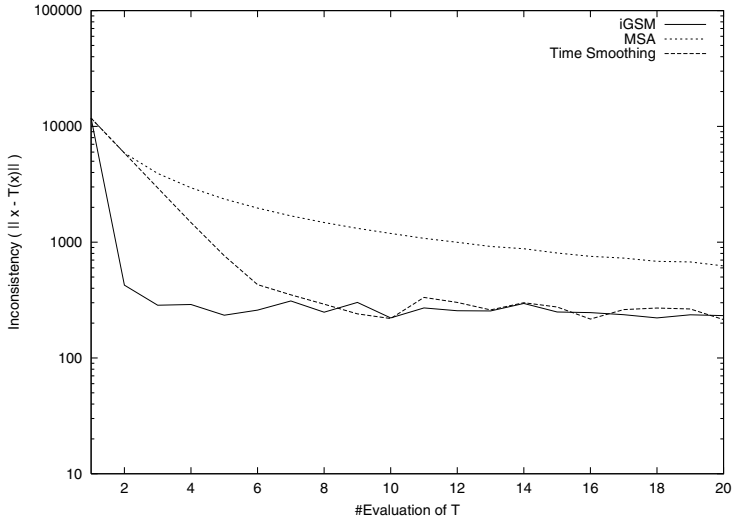


Figure 6.28: Irvine network (x_0^B) 5:00am-5:45am

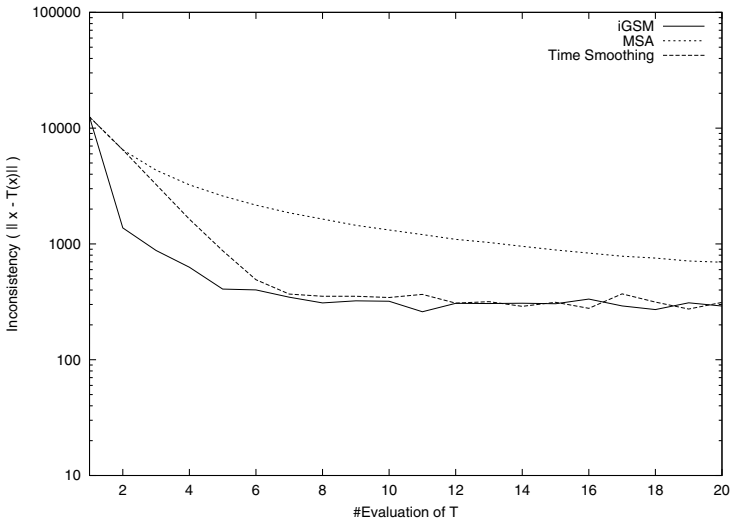


Figure 6.29: Irvine network (x_0^B) 5:15am-6:00am

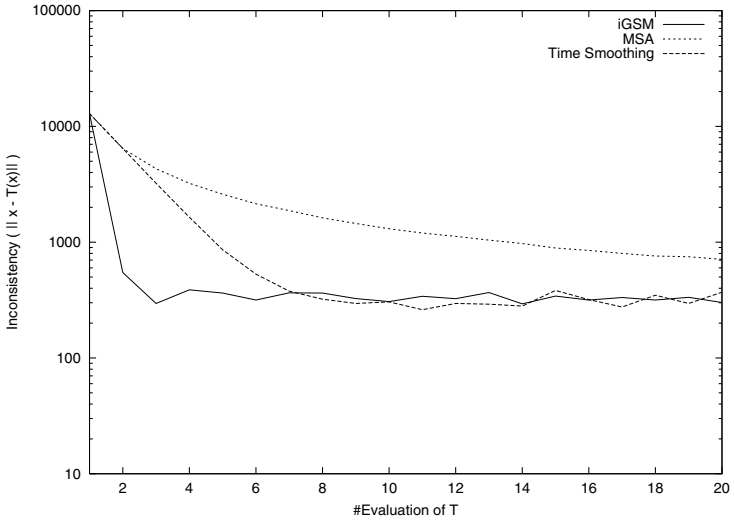


Figure 6.30: Irvine network (x_0^B) 5:30am-6:15am

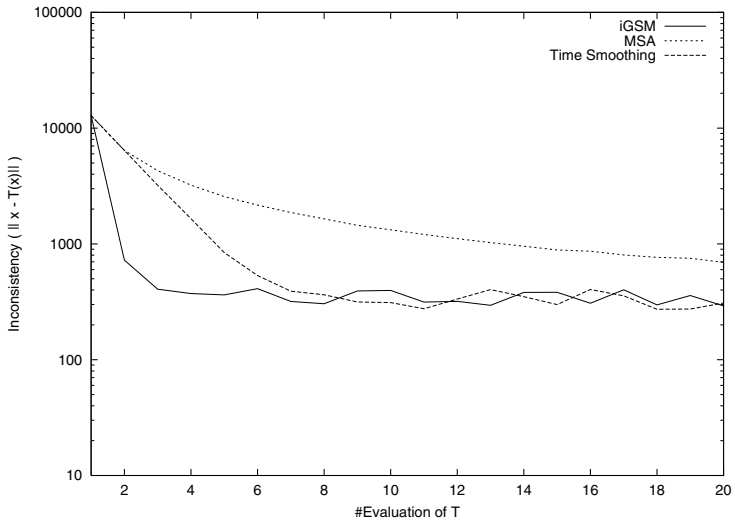


Figure 6.31: Irvine network (x_0^B) 5:45am-6:30am

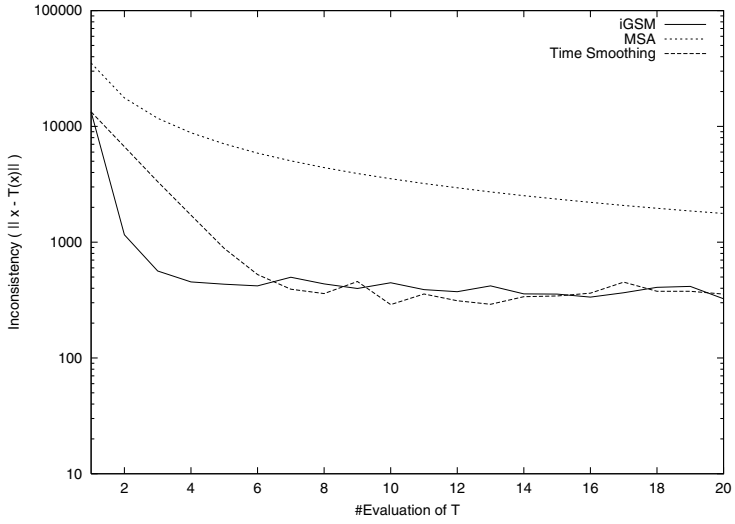


Figure 6.32: Irvine network (x_0^B) 6:00am-6:45am

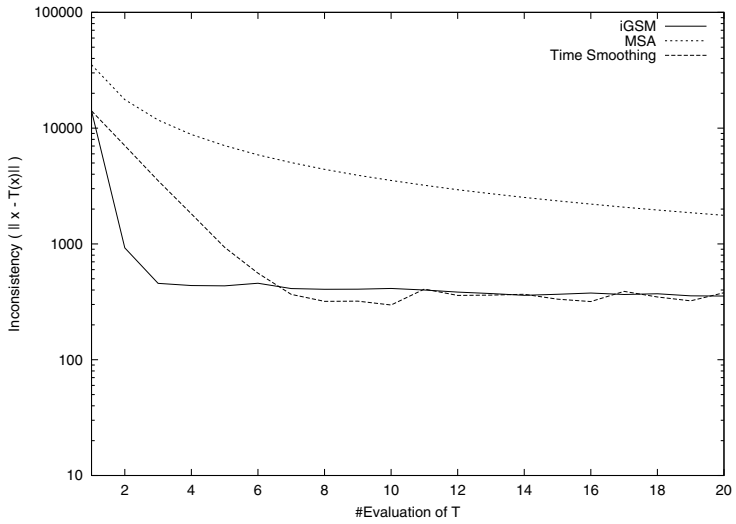


Figure 6.33: Irvine network (x_0^B) 6:15am-7:00am

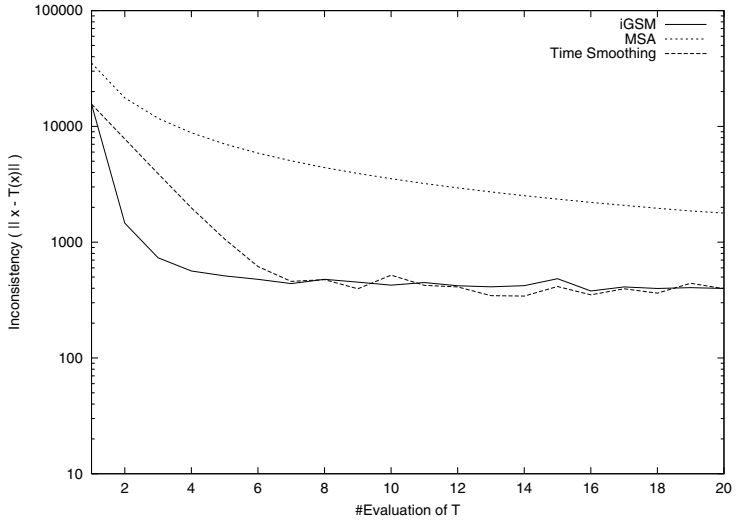


Figure 6.34: Irvine network (x_0^B) 6:30am-7:15am

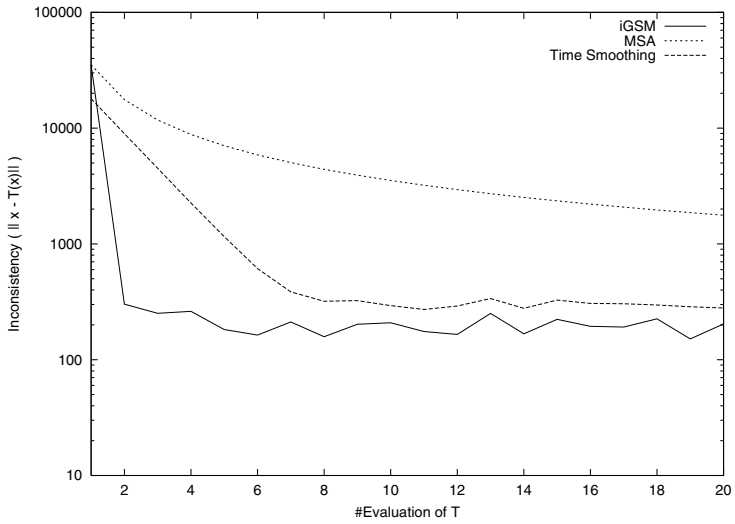


Figure 6.35: Irvine network (x_0^B) 6:45am-7:30am

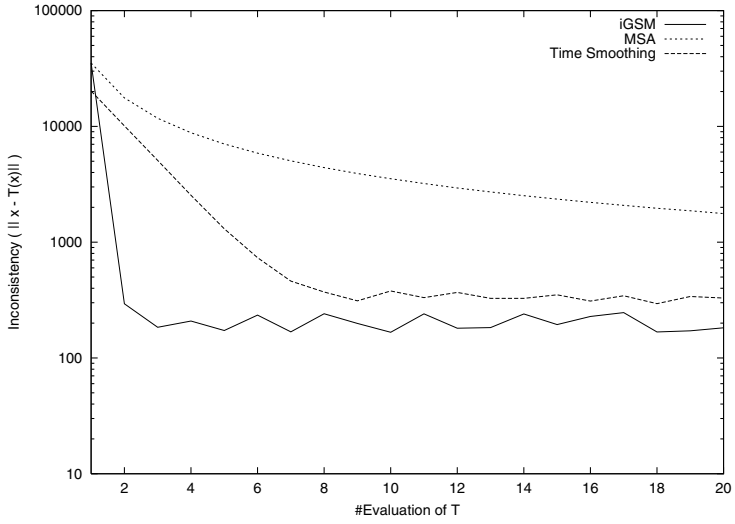


Figure 6.36: Irvine network (x_0^B) 7:00am-7:45am

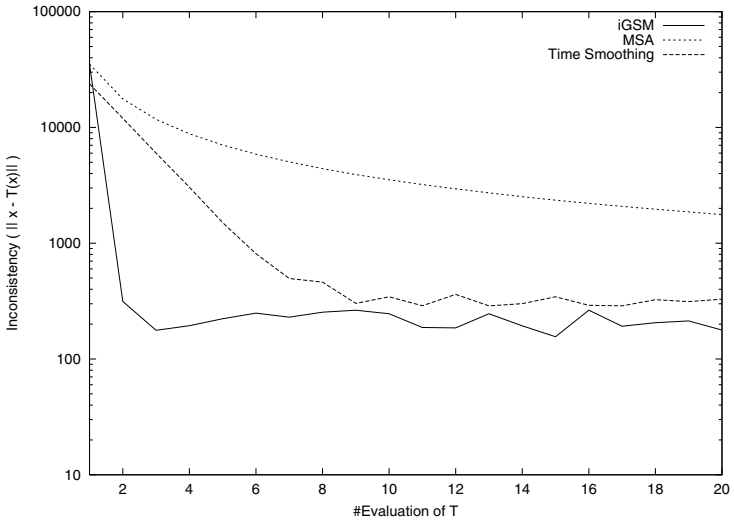


Figure 6.37: Irvine network (x_0^B) 7:15am-8:00am

Chapter 7

Conclusion

7.1 Review of key results

This dissertation proposes mainly two new algorithmic approaches for solving basic numerical problems encountered in computational science:

- a new approach for solving large-scale incremental least squares,
- a new class of methods for solving systems of nonlinear equations.

These new methods have then been applied to two real-time problems arising in the context of Intelligent Transportation Systems:

- the estimation and prediction of Origin-Destination tables,
- the generation of consistent anticipatory route guidance.

The principal objective of a method solving a large-scale real-time problem is to be able to compute a "good" solution in a tight given amount of time, sometimes expressed by a computation budget. In this dissertation several new results have been proposed representing progress towards this general goal. For both problems the same process has been applied. First we have reformulated the problems taking into account their own characteristics in a real-time context and showed their equivalence with the original problems. Second new methods have been proposed to solve both problems using these new formulations.

The OD estimation and prediction problem is classically formulated and solved using a Kalman filter approach, which refers to both formulation and resolution algorithm. This is an incremental method solving least-squares allowing to update the solution when additional data is available. Furthermore it is not designed to solve large-scale problems. In Chapter 3 a more classical least-squares modeling approach, inspired by formulations proposed by Cascetta et al. (1993) and Ashok and Ben-Akiva (1993), has been considered. The equivalence between the classical Kalman filter formulation and this approach has been theoretically demonstrated.

This new formulation allows to consider efficient methods of resolution. We have given preference to the LSQR algorithm, proposed by Paige and Saunders (1982). This conjugate gradient method is designed to efficiently solve large-scale least squares exploiting the sparse structure of the problem. Unfortunately this algorithm is not incremental and, therefore not designed to solve real-time applications. Therefore we had to impose a simplification of the formulation. It has been shown that this simplification does not deteriorate the quality of the solution of the original problem. Numerical results computed on real data emphasize the potential of this new approach for real-time problems in presence of sparsity as it significantly decreases the computation time for the estimation and the prediction of OD tables compared to the Kalman filter approach.

While this method is motivated in this dissertation by a particular application in the transportation field, the proposed approach is general and applicable to any incremental least-squares problem.

The consistent anticipatory route guidance, described in Chapter 6, is generally formulated as a fixed point problem. This problem has to be solved quickly and accurately enough for the results to be timely and of use to drivers, despite the considerable amounts of computation involved in the guidance generation. In this context the performances, in terms of speed of convergence, of classical fixed point algorithms are insufficient. In this dissertation we have reformulated the CARG problem as the resolution of a system of nonlinear equations, which is totally equivalent to a fixed point formulation. The unavailability of analytical forms for the objective function and the presence of noise prevent from using state-of-the-art methods designed to solve large-scale systems of nonlinear equations, given that most of them are based on finite differences approaches.

A new class of iterative methods has been introduced. The main idea is to generalize classical secant methods by building the secant model using more than two previous iterates. This approach allows to use all available information collected throughout the iterations to construct the model. This class of methods can be considered as the first efficient multi-iterates generalization of secant methods. We also show that our approach can lead to an update formula. In that case, we prove the q-linear convergence of the corresponding quasi-Newton method. Numerically these methods outperform classical quasi-Newton methods in terms of robustness and efficiency.

We derive from this class of methods a matrix-free algorithm designed to solve large-scale systems of nonlinear equations. Practically this method allows us to solve very large size problems with more than one million variables. Computational experiments on standard problems show that this algorithm outperforms classical large-scale quasi-Newton methods in terms of efficiency and robustness. Its numerical performances can even be compared to Newton-Krylov methods currently considered as the best to solve large-scale systems of nonlinear equations. To our best knowledge iGSM is the first matrix-free algorithm solving systems of nonlinear equations without resorting to finite

differences. Moreover numerical evidence of the superiority of our new class of methods for solving noisy system of nonlinear equations compared to classical large-scale methods has been given.

Considering the CARG problem as the resolution of a system of nonlinear equations permit to use new resolution algorithms. Two system softwares have been used to produce the numerical results. They emphasize the good behavior of iGSM especially in a real-time context, as it significantly decreases the value of the inconsistency in the first iterates compared to classical methods to solve fixed point problems. This numerical study has also highlighted some interesting points about the CARG problem outlined in the next section.

7.2 Future research

There are many potential paths that future research may take in continuation of this work. Some of these were brought to light by this dissertation, and others are peripherally related. The most promising of these paths are outlined in this section.

Nonlinear programming issues

This thesis has brought novelties in the field of resolution of large-scale systems of nonlinear equations. We hope that these ideas can be tailored to optimization problems.

- More precisely, as described in Chapter 4, GSM uses a least-squares approach to calibrate, at each iteration, an approximation of the Jacobian based on previous iterates. This approach has proven to be numerically very efficient, in particular in presence of noise and for large-scale problems. We are convinced that exploiting this idea for unconstrained optimization could produce interesting advances.

  Consider the following problem:

  $$\min_{x} f(x) \tag{7.1}$$

  where $x \in \mathbb{R}^n$ and where $f : \mathbb{R}^n \to \mathbb{R}$. Most of algorithms designed to solve (7.1) are iterative methods starting from an initial point x_0 and stopping when a solution has been found with a predetermined accuracy (cf Bierlaire and Crittin, 1999 for a review of these methods). These algorithms basically try to approximate the objective function by suitable models which are progressively built and update during the iterations of the algorithm. At each iteration the model is minimized to obtain the next iterate. Different strategies can be considered to built such models.

  The first class of methods are based on collecting derivative information about the objective function to construct a linear or quadratic model.

These models can be simply estimated using the analytical derivatives. However the analytical derivatives of the objective function are often unavailable or expensive to compute. That is why in practice we prefer to evaluate the derivatives numerically using finite differences techniques or algorithmic differentiation (Griewank, 1994, Griewank, 2000). With these procedures the computational cost of evaluating the Hessian remains often expensive, and quasi-Newton approaches are preferred to approximate it (Dennis and Schnabel, 1996). Preliminary numerical results, that we have conducted on classical problems where the quasi-Newton approximation of the Hessian has been substituted by a GSM-type of approximation have shown a great potential.

A second type of approaches is based on the idea of modeling the objective function using interpolation of the previous iterates, instead of modeling its derivatives. The first ideas for interpolation methods seem to have been proposed by Winfield (1969). The work by Powell (1970) is usually considered as seminal in this field. The idea is to replace the expensive objective function by a quadratic model that interpolates the function at specific points. The main difficulty is to maintain good geometrical properties of the interpolation points (De Boor and Ron, 1992, Powell, 1994). An efficient trust-region based algorithm (DFO) has been proposed by Conn and Toint (1995), and the convergence properties have been analyzed by Conn et al. (1997). This algorithm has been shown to perform well with practical applications (Booker et al., 1998). A variant of DFO based on Lanczos method has been proposed by Gould et al. (1999). Unfortunately, these methods are designed for small-scale problems. Indeed, $(n+1)(n+2)/2$ interpolation points are necessary to compute a model of the objective function. Recently Powell (2003) has proposed a method that updates the quadratic model by minimizing the Frobenius norm of the change of the Hessian. This approach allows him to decrease the number of interpolation points. Oeuvray and Bierlaire (2003) suggest to use radial basis functions to interpolate the objective function instead of classical polynomials models. It constitutes another alternative to reduce the number of interpolation points. Following the ideas of Chapter 4 we propose to replace the exact interpolation used by these methods by a least-squares approach to construct the model. It will allow to construct a model using all the previous iterates without particular assumptions on the geometry or the number of the points necessary for the construction of the model.

Moreover the generalization of GSM to unconstrained optimization, as for the resolution of systems of nonlinear equations, will allow to consider noisy optimization problems, which is impracticable for both interpolation and derivative-based methods. Moreover we hope that the matrix-free approach proposed in Chapter 5 can be transposed to optimization.

It would produce a new kind of efficient optimization methods especially designed to solve large-scale problems.

Although the most straightforward extension of GSM has been suggested for unconstrained optimization, we are convinced that our idea to calibrate a second order model using a least-squares approach can be applied with success in any domain where a model has to be computed, especially in constrained optimization.

The extension of the methods proposed in this dissertation to the optimization field is an exciting and promising project.

Transportation issues

In this dissertation the transportation problems have been used as practical applications to test the algorithmic developments presented. In the light of numerical results obtained some interesting doors have been opened.

- As described in Chapter 2 the OD estimation and prediction process involves assignment matrices, which represent the mapping of OD flows into link flows computed by the supply simulator, which itself uses OD matrices as inputs. Therefore the OD estimation and prediction can be formulated as a fixed point problem. Generally, for example in DynaMIT, this fixed point is computed using the functional iteration method. Encouraged by the numerical results obtained for the guidance generation, we anticipate that a more efficient algorithm to compute the fixed point, like iGSM, could noticeably speed up the process of estimation and prediction of OD tables.

- The influence of the starting point on the efficiency of algorithms used to solve the CARG problem has to be analyzed more deeply. The generation of route guidance is performed frequently, on the same network, but with traffic conditions that vary in time. In most cases, when the traffic conditions do not vary drastically, the solution of previously solved instances of the problem may be a good approximation of the solution of the next instance. But astonishingly in Chapter 6 we have shown that a "good" starting point, i.e. with a low value of inconsistency, not always produces a more consistent guidance than a "bad" one. Of course it is dependent on the resolution algorithm used to solve the associated fixed point problem, but this could have far reaching consequences on real-time applications. Numerical studies comparing the behavior of algorithms with different starting points, selected either close to the solution, or arbitrarily far from it, would produce a better understanding of its influence.

- Additional characterizations of consistent anticipatory route guidance seem necessary in a real-time context. Indeed, preliminary results indicate

that two guidances with the same level of inconsistency may produce different future traffic conditions. We have to be able to choose between different available guidances for a given time horizon. Perhaps a reasonable place to start is to consider other formulations of the CARG, including the two other composite maps define by (6.3) and (6.4) that characterize the fixed point formulation. The path time variable appeared to involve the least amount of stochasticity, as described by Bottom (2000), and will probably produce a good indicator for the quality of the associated guidance.

Despite the fact that the applications described in this thesis were essentially Intelligent Transportation Systems, we are persuaded that the presented approaches have a wide range of other applications in many domains.

An alternative to alleviate traffic jam problems

"It's very good jam," said the Queen.
"Well, I don't want any to-day, at any rate."
"You couldn't have it if you did want it," the Queen said. "The rule is jam tomorrow and jam yesterday but never jam to-day."
"It must come sometimes to 'jam to-day' " Alice objected.
"No it can't," said the Queen. "It's jam every other day; to-day isn't any other day, you know."
"I don't understand you," said Alice. "It's dreadfully confusing."

Lewis Caroll, *Alice's Adventures in Wonderland*, 1865.

Bibliography

Antoniou, C., Ben-Akiva, M., Bierlaire, M. and Mishalani, R. (1997). Demand simulation for dynamic traffic assignment, *Proceedings of the 8th IFAC Symposium on Transportation Systems*, Chania, Greece.

Arnott, R., de Palma, A. and Lindsey, R. (1991). Does providing information to drivers reduce traffic congestion?, *Transportation Research A* **25**(5): 309–318.

Ascher, U. and Russel, R. D. (1985). Numerical boundary value ODEs, Boston Birkhauser.

Ashok, K. (1996). *Estimation and Prediction of Time-Dependent Origin-Destination Flows*, PhD thesis, Massachusetts Institute of Technology, Cambridge, Ma.

Ashok, K. and Ben-Akiva, M. (1993). Dynamic origin-destination matrix estimation and prediction for real-time traffic management systems, *in* C. Daganzo (ed.), *Transportation and Traffic Theory*, Elsevier Science Publishing Company Inc. Proceedings of the 12th ISTTT.

Ashok, K. and Ben-Akiva, M. (2000). Alternative approaches for real-time estimation and prediction of time-dependent origin-destination flows, *Transportation Science* **34**(1): 21–36.

Banach, S. (1922). Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales, *Fund. Math.* **3**: 133–181.

Barceló, J. and Casas, J. (1999). The use of neural networks for short-term prediction of traffic demand, *in* A. Ceder (ed.), *Transportation and Traffic Theory. Proceedings of the 14th ISTTT*, Pergamon, pp. 419–443.

Ben-Akiva, M. and Bierlaire, M. (1999). Discrete choice methods and their applications to short-term travel decisions, *in* R. Hall (ed.), *Handbook of Transportation Science*, Kluwer, pp. 5–34.

Ben-Akiva, M., Bierlaire, M., Bottom, J., Koutsopoulos, H. N. and Mishalani, R. G. (1997). Development of a route guidance generation system for real-time application, *Proceedings of the 8th IFAC Symposium on Transportation Systems*, Chania, Greece.

Ben-Akiva, M., Bierlaire, M., Burton, D., Koutsopoulos, H. and Mishalani, R. (2001). Network state estimation and prediction for real-time transportation management applications, *Networks and Spatial Economics* **1**(3/4): 293–318.

Ben-Akiva, M., Bierlaire, M., Koutsopoulos, H. N. and Mishalani, R. (2002). Real-time simulation of traffic demand-supply interactions within DynaMIT, *in* M. Gendreau and P. Marcotte (eds), *Transportation and network analysis: current trends. Miscellenea in honor of Michael Florian*, Kluwer Academic Publishers, Boston/Dordrecht/London.

Ben-Akiva, M. E., Bierlaire, M., Koutsopoulos, H. N. and Mishalani, R. G. (1998). DynaMIT: a simulation-based system for traffic prediction, *Proceedings of the DACCORD Short-Term forecasting workshop*.

Ben-Akiva, M. E., de Palma, A. and Kaysi, I. (1996). The impact of predictive information on guidance efficiency: an analytical approach, *in* L. Bianco and P. Toth (eds), *Advanced Methods in Transportation Analysis*, Springer-Verlag, pp. 413–432.

Bertsekas, D. P. (1995). *Nonlinear Programming*, Athena Scientific, Belmont.

Bierlaire, M. and Crittin, F. (1999). Optimisation de grande taille sans dérivée: Etat de l'art et perspectives, *Technical Report RO-1999.0404*, Swiss Insitute of Technology, Lausanne, ROSO-DMA-EPFL CH-1015 Lausanne.

Bierlaire, M. and Crittin, F. (2001). New algorithmic approaches for the anticipatory guidance generation problem, *Proceedings of the 1st Swiss Transportation Research Conference*, Ascona, Switzerland.

Bierlaire, M. and Crittin, F. (2003a). Generalization of secant methods for solving systems of nonlinear equations, *Proceedings of the 3rd Swiss Transportation Research Conference*, Ascona, Switzerland.

Bierlaire, M. and Crittin, F. (2003b). Solving large-scale systems of nonlinear equations, *Technical Report RO-2003.0804*, Swiss Insitute of Technology, Lausanne, ROSO-DMA-EPFL CH-1015 Lausanne. Submitted to Mathematical Programming.

Bierlaire, M. and Crittin, F. (2003c). Solving the anticipatory route guidance generation problem using a generalization of secant method, *Proceedings of the 3rd Swiss Transportation Research Conference*, Ascona, Switzerland.

Bierlaire, M. and Crittin, F. (2004). An efficient algorithm for real-time estimation and prediction of dynamic OD tables, *Operations Research* ,**Forthcoming**. Also available as Report RO-010808, ROSO-DMA-EPFL CH-1015 Lausanne.

Bierlaire, M., Mishalani, R. and Ben-Akiva, M. (2000). General framework for dynamic demand simulation, *Technical Report RO-000223*, ROSO-DMA-EPFL Swiss Institute of Technology, CH-1015 Lausanne.

Bierlaire, M. and Toint, P. L. (1995). MEUSE: an origin-destination estimator that exploits structure, *Transportation Research B* **29**(1): 47–60.

Bierlaire, M., Toint, P. L. and Tuyttens, D. (1991). On iterative algorithms for linear least squares problems with bound constraints, *Linear Algebra and its Applications* **143**: 111–143.

Blum, J. R. (1954). Multidimensional stochastic approximation methods, *Annals of mathematical statistics* **25**(4): 737–744.

Bogle, I. and Perkins, J. (1990). A new sparsity preserving quasi-Newton update for solving nonlinear equations, *SIAM Journal on scientific and statistical computations* **11**: 621–630.

Booker, A. J., Dennis, J. E., Frank, P. D., Serafini, D. B., Torczon, V. and Trosset, M. W. (1998). A rigourous framework for optimization of expensive functions by surrogates, *Technical report*, Mathematics & Engineering Analysis, Boeing Shared Services Group.

Bottom, J. (2000). *Consistent Anticipatory Route Guidance*, PhD thesis, Massachusetts Institute of Technology.

Bottom, J., Ben-Akiva, M., Bierlaire, M., Chabini, I., Koutsopoulos, H. and Yang, Q. (1999). Investigation of route guidance generation issues by simulation with DynaMIT, *in* A. Ceder (ed.), *Transportation and Traffic Theory. Proceedings of the 14th ISTTT*, Pergamon, pp. 577–600.

Broyden, C. G. (1965). A class of methods for solving nonlinear simultaneous equations, *Mathematics of Computation* **19**: 577–593.

Broyden, C. G., Dennis, J. E. and Moré, J. J. (1973). On the local and superlinear convergence of quasi-Newton methods, *Journal of the Institute of Mathematics and its Applications* **12**: 233–246.

Byrd, R., Nocedal, J. and Schnabel, R. B. (1994). Representation of quasi-Newton matrices and their use in limited memory methods, *Mathematical Programming* **63**: 129–136.

Cantarella, G. E. (1997). A general fixed-point approach to multimode multi-user equilibrium assignement with elastic demand, *Transportation Science* **31**(2): 107–128.

Cascetta, E., Inaudi, D. and Marquis, G. (1993). Dynamic estimators of origin-destination matrices using traffic counts, *Transportation Science* **27**(4): 363–373.

Casey, H. J. (1955). Applications to traffic engineering of the law of retail gravitation, *Traffic Quaterly* **9**(1): 23–35.

Chang, G. L. and Wu, J. (1994). Recursive estimation of time-varying OD flows from traffic counts in freeway corridors, *Transportation Research B* **28**.

Choi, T. D. and Kelley, C. (2000). Superlinear convergence and implicit filtering, *Siam Journal of Optimization* **10**.

Chui, C. and Chen, G. (1991). *Kalman Filtering with Real-Time Applications*, Spinger-Verlag.

Chung, K. L. (1954). On a stochastic approximation method, *Annals of mathematical statistics* **25**(3): 463–483.

Conn, A. R. and Toint, P. L. (1995). An algorithm using quadratic interpolation for unconstrained derivative free optimization, *in* G. D. Pillo and F. Gianessi (eds), *Nonlinear Optimization and Applications*, Plenum Publishing.

Conn, A., Scheinberg, K. and Toint, P. L. (1997). On the convergence of derivative-free methods for unconstrained optimization, *in* A. Iserles and M. Buhmann (eds), *Approximation Theory and Optimization: Tributes to M.J.D. Powell*, Cambridge University Press, Cambridge, UK, pp. 83–108.

De Boor, C. and Ron, A. (1992). Computational aspects of polynomial interpolation in several variables, *Mathematics of Computation* **58**(198): 705–727.

Dennis, J. E. and Schnabel, R. B. (1996). *Numerical methods for unconstrained optimization and nonlinear equations*, Society for Industrial and Applied Mathematics.

Dolan, E. and More, J. (2002). Benchmarking optimization software with performance profiles, *Mathematical Programming, Serie A* **91**: 2.

Dunn, J. C. (1973). On recursive averaging processes and Hilbert space extensions of the contractions mapping principle, *Journal of Franklin Institute* **295**: 93–116.

Eaton, J. W. (1997). GNU Octave: a high level interactive language for numerical computations, `www.octave.org`.

Engelson, L. (1997). Self-fulfilling and recursive forecasts — an analytical perspective for driver information systems, *Proceedings of 8th IATBR meeting*. Austin, Texas.

Fabian, V. (1973). Asymptotically efficient stochastic approximations: the RM case, *Annals of statistics* **1**(3): 486–495.

Florian, M. and Chen, Y. (1993). A coordinate descent method for the bilevel OD matrix adjustment problem, *IFORS conference*, Lisbon.

Ford, J. A. (1999). A survey of multi-step quasi-newton methods, *Proceedings of the International Conference on Scientific Computations*, Beirut, Lebanon.

Ford, J. and Moghrabi, I. (1997). Alternating multi-step quasi-newton methods for unconstrained optimization, *Journal of Computational and Applied Mathematics* **82**: 105–116.

Friedlander, A., Gomes-Ruggiero, M. A., Kozakevich, D. N., Martnez, J. M. and Santos, S. A. (1997). Solving nonlinear systems of equations by means of quasi-newton methods with a nonmonotone strategy, *Optimization methods and Software* **8**: 25–51.

Golub, G. H. and Van Loan, C. F. (1996). *Matrix Computations*, third edn, Johns Hopkins University Press, Baltimore and London.

Gomes-Ruggiero, M. A., Martinez, J. M. and Moretti, A. C. (1991). Comparing algorithms for solving sparse systems of nonlinear equations, *Siam J. Sci. Stat Comp.* .

Gould, N., Lucidi, S., Roma, M. and Toint, P. L. (1999). Solving the trust region subproblem with the Lanczos method, *SIAM Journal on Optimization* .

Gragg, W. and Stewart, G. (1976). A stable variant of the secant method for solving nonlinear equations, *SIAM Journal on Numerical Analysis* **13**: 889–903.

Griewank, A. (1994). Computational differentiation and optimization, *in* J. R. Birge and K. G. Murty (eds), *Mathematical Programming: State of the Art 1994*, The University of Michigan, Ann Arbour, USA, pp. 102–131.

Griewank, A. (2000). *Evaluation derivatives: Principles and Techniques of Algorithmic differentiation*, Frontiers in Applied Mathematics, Siam.

Hall, R. W. (1986). The fastest path through a network with random time-dependent travel times, *Transportation Science* **20**(3): 182–188.

Ishikawa, S. (1976). Fixed points and iteration of a nonexpansive mapping in Banach space, *Proceedings of the American Mathematical Society* **59**: 65–71.

Kalman, R. E. (1960). A new approach to linear filtering and prediction problems, *J. of Basic Eng., Trans. ASME, Series D* **82**(1): 33–45.

Kaufman, D. E., Smith, R. L. and Wunderlich, K. E. (1998). User-equilibrium properties of fixed points in iterative dynamic routing/assignment methods, *Transportation Research C* **6**: 1–16.

Kaysi, I. (1992). *Framework and Models for the Provision of Real-Time Driver Information*, PhD thesis, Massachusetts Institute of Technology, Cambridge, Ma.

Kelley, C. T. (1995). *Iterative methods for linear and nonlinear equations*, Frontiers in Applied Mathematics, SIAM, Philadephia, Pennsylvania.

Kelley, C. T. (2002a). Private communication.

Kelley, C. T. (2002b). *Solving nonlinear equations with Newton's method*, Draft stage. http://www4.ncsu.edu/~ctk/newton.html.

Kushner, H. J. and Yang, J. (1993). Stochastic approximation with averaging of the iterates: optimal asymptotic rate of convergence for general process, *SICON* **31**(4): 1045–1062.

Li, G. (1989). Successive column correction algorithms for solving sparse systems of nonlinear equations, *Mathematical Programming* **43**: 187–207.

Luksan, L. and Vlcek, J. (1998). Computational experience with globally convergent descent methods for large sparse systems of nonlinear equations, *Optimization Methods and Software* **8**: 185–199.

Magnanti, T. L. and Perakis, G. (1994). Averaging schemes for solving fixed point and variational inequality problems, *Working paper OR 296-94*, Operation Research Center, Massachusetts Institute of Technology, Bldg. E40-149, 77 Massachusetts Avenue, Cambridge, MA 02139-4307.

Magnanti, T. and Perakis, G. (1997). Averaging schemes for variational inequalities and systems of equations, *Mathematics of operations research* **22**(3): 568–587.

Mahmassani, H., Hu, T., Peeta, S. and Ziliaskopoulos, A. (1993). Development and testing of dynamic traffic assignment and simulation procedures for ATIS/ATMS applications, *Technical Report DTFH61-90-R-00074-FG*, Center for Transportation Research, University of Texas at Austin.

Martinez, J. M. (1979). Three new algorithms based on the sequential secant method, *BIT* **19**: 236–243.

Martinez, J. M. (1982). Sobre dois metodos de broyden, *Matematica Aplicada e Computational* **1**.

Martinez, J. M. (1984). A quasi-newton method with modification of one column per iteration, *Computing* **33**.

Martinez, J. M. (2000). Practical quasi-newton methods for solving nonlinear systems, *Journal of Computational and Applied Mathematics* **124**: 97–122.

Martinez, J. M. and Zambaldi, M. C. (1992). An inverse column-updating method for solving large-scale systems of nonlinear equations, *Optimization method and software* **1**.

Moghrabi, I. (1993). *Multi-step quasi-Newton methods for optimization*, PhD thesis, University of Essex, United Kingdom.

Moré, J. J., Garbow, B. S. and Hillstrom, K. E. (1981). Testing unconstrained optimization software, *ACM Transactions on Mathematical Software* **7**(1): 17–41.

Nagel, K. (1997). Experiences with iterated traffic microsimulations in Dallas, *in* M. Schrekenberg and D. E. Wolf (eds), *Proceedings of the workshop on traffic and granular flow*, SV, Duisburg, Germany.

Nocedal, J. and Wright, S. J. (1999). *Numerical optimization*, Operations Research, Springer Verlag.

Oeuvray, R. and Bierlaire, M. (2003). Boosters: A derivative-free algorithm based on radial basis functions, *Technical report RO-2003.0918*, Operation Research Group (ROSO) - IMA - EPFL.

Okutani, I. (1987). The Kalman filtering approaches in some transportation and traffic problems, *in* N. H. Gartner and N. H. M. Wilson (eds), *Transportation and traffic theory*, Elsevier, New-York, pp. 397–416. Proceedings of the Tenth International Symposium on Transportation and Traffic Theory, July 8–10 1987, MIT, Cambridge, Massachusetts.

Okutani, I. and Stephanades, Y. (1984). Dynamic prediction of traffic volume through Kalman filtering theory, *Transportation Research* **18**.

Ortega, J. M. and Rheinboldt, W. C. (1970). *Iterative solution of nonlinear equations in several variables*, Academic Press, New York.

Paige, C. C. and Saunders, M. A. (1982). LSQR: an algorithm for sparse linear equations and sparse least squares, *ACM Transactions on Mathematical Software* **8**: 43–71.

Polyak, B. T. and Juditsky, A. B. (1992). Acceleration of stochastic approximation by averaging, *SICON* **30**(4): 838–855.

Powell, M. J. D. (1970). A new algorithm for unconstrained optimization, *in* J. B. Rosen, O. L. Mangasarian and K. Ritter (eds), *Nonlinear Programming*, Academic Press, New York.

Powell, M. J. D. (1994). A direct search optimization method that models the objective by quadratic interpolation, Presentation at the 5th Stockholm Optimization Days.

Powell, M. J. D. (2003). On the use of quadratic models in unconstrained minimization without derivatives, *Technical report*, Departement of applied mathematics and theoretical physics,Cambridge University.

Pozzo, R. (1998). Template Numerical Toolkit (TNT), an interface for scientific computing in C++, `http://math.nist.gov/tnt/`.

Robbins, H. and Monro, S. (1951). A stochastic approximation method, *Annals of mathematical statistics* **22**(3): 400–407.

Robert, S. M. and Shipman, J. S. (1976). On the closed form solution of Troesch's problem, *J. Comput. Physics* .

Roose, A., Kulla, V., Lomp, M. and Meressoo, T. (1990). Test examples of systems of non-linear equations, *Estonian software and Computer Service Company* .

Ruggiero, M. G., D.N.Kozakevich and J.M.Martnez (1996). A numerical study on large-scale nonlinear solvers, *Computers and Mathematics with Applications: An International Journal* **32**: 1–13.

Saad, Y. and Schultz, M. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems, *SIAM Journal on Numerical Analysis* **7**: 856–869.

Schnabel, R. B. and Eskow, E. (1991). A new modified Cholesky factorization, *SIAM Journal on Scientific and Statistical Computing* **11**: 1136–1158.

Sheffi, Y. and Powell, W. (1982). An algorithm for the equilibrium assignment problem with random link times, *Networks* **12**: 191–207.

Smith, B. L., Pack, M. L., Lovell, D. J. and Sermons, M. W. (2001). Transportation management applications of anonymous mobile call sampling, *Proceedings of the 11th Annual Meeting of ITS America, Miami, FL.*

Spedicato, E. and Huang, Z. (1997). Numerical experience with newton-like methods for nonlinear algebraic systems, *Computing* **58**: 69–99.

The Mathworks Inc. (1994). Matlab optimization toolbox, The Mathworks Inc.

Toint, P. L. (1986). Numerical solution of large sets of algebraic nonlinear equations, *Mathematics of Computation* **46**(173): 175–189.

Transportation Research Board (1999). *Research on intelligent transportation systems, human factors, and advanced traveler information system design and effects*, National Academy Press, National Research Council. Washington, D.C.

van der Zijpp, N. (1996). *Dynamic Origin-Destination Matrix Estimation on Motorway Networks.*, PhD thesis, Dpt of Civil Rnginnering, Delft University of Technology.

van der Zijpp, N. and Hammerslag, R. (1994). Improved Kalman filtering approach for estimating origin-destination matrices for freeway corridors, *Transportation Research Record* **1443**.

Van Zuylen, H. J. and Willumsen, L. G. (1980). The most likely trip matrix estimated from traffic counts, *Transportation Research B* **14**: 281–293.

Willumsen, L. G. (1981). *An entropy maximising model for estimating trip matrices from traffic counts*, PhD thesis, University of Leeds.

Wilson, A. G. (1970). *Entropy in urban and regional modelling*, Pion, London.

Winfield, D. (1969). *Function and functional otpimization by interpolation in a data table*, PhD thesis, Harvard University, Cambridge, MA.

Wolfe, P. (1959). The secant method for solving nonlinear equations, *Communications ACM* **12**: 12–13.

Frank Crittin

http://rosowww.epfl.ch/fc
frank.crittin@epfl.ch

Ch. des Clochetons 7
1004 Lausanne
Born October 13, 1972

EDUCATION

1992 – **École Polytechnique Fédérale de Lausanne (EPFL)**:

2003: Postgraduate course "*Discrete Choice Analysis: Predicting Demand and Market Share*".
1999: Postgraduate course "*Operations Research and Statistics*".
1998: Diploma in mathematical engineering.

1987 – 1992 **Secondary school at the Lycée Collège des Creusets, Sion**: Maturity C (scientific).

PROFESSIONAL EXPERIENCE

1998 – **EPFL – Institute of Mathematics, Lausanne**:

Research assistant of Prof. Th. M. Liebling and Dr M. Bierlaire, chair of Operations Research. Supervision of semester and diploma projects.

1999 – **IML – Institut International de Management pour la Logistique**:

Lecturer for IML (Module M7: Modeling and Simulation in Logistics.)

VARIOUS

Languages: Fluent in French and English, good knowledge of German, notions of Italian.

Radio presenter on *Radio Rhône*, Sion (2000).

Volleyball and tennis player.